



**Technische Universität Ilmenau**

Fakultät für Informatik und Automatisierung

Fachgebiet Neuroinformatik und Kognitive Robotik

# **Klassifikation von Berührungsmustern für einen textilen haptischen Sensor**

Bachelorarbeit zur Erlangung des akademischen Grades Bachelor of Science

**David Kudlek**

Betreuer: Dipl.-Inf. Steffen Müller

Verantwortlicher Hochschullehrer:

Prof. Dr. H.-M. Groß, FG Neuroinformatik und Kognitive Robotik

Die Bachelorarbeit wurde am 01.10.2014 bei der Fakultät für Informatik  
und Automatisierung der Technischen Universität Ilmenau eingereicht.

[urn:nbn:de:gbv:ilm1-2014200172](https://nbn-resolving.org/urn:nbn:de:gbv:ilm1-2014200172)





Erklärung: „Hiermit versichere ich, dass ich diese Bachelorarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Alle von mir aus anderen Veröffentlichungen übernommenen Passagen sind als solche gekennzeichnet.“

Ilmenau, 01.10.2014

.....  
David Kudlek

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>State of the Art</b>	<b>3</b>
2.1	Allgemeine Einführung . . . . .	3
2.1.1	Sensortypen . . . . .	3
2.1.2	Einsatzgebiete . . . . .	4
2.2	Textile Sensoren . . . . .	5
2.2.1	Ganzkörpersensor . . . . .	5
2.2.2	Sensorhandschuh . . . . .	6
2.2.3	Sensorfläche . . . . .	7
2.2.4	Bewertung Sensoren . . . . .	8
2.3	Klassifikation von Berührungen . . . . .	11
2.3.1	Nächste-Nachbarn-Klassifikation (KNN) . . . . .	11
2.3.2	Neuronales Netz (NN) . . . . .	12
2.3.3	Support Vector Machine(SVM) . . . . .	14
2.3.4	Bewertung Klassifikatoren . . . . .	16
2.4	Auswertung der Recherche . . . . .	18
<b>3</b>	<b>Vom Fachgebiet entwickelte Sensoren</b>	<b>19</b>
3.1	Textiler Sensor mit leitfähigem Garn . . . . .	20
3.1.1	Technischer Aufbau . . . . .	20
3.1.2	Messprinzip Variante 1 . . . . .	20
3.1.3	Messprinzip Variante 2 . . . . .	26

---

3.2	Textiler Sensor mit silberbeschichteten Stoffen . . . . .	29
3.2.1	Technischer Aufbau . . . . .	29
3.2.2	Messprinzip . . . . .	32
<b>4</b>	<b>Merkmalsextraktion und Klassifikation</b>	<b>35</b>
4.1	Ansatz mit Zeitfenster . . . . .	36
4.1.1	Merkmalsextraktion . . . . .	36
4.1.2	Klassifikation . . . . .	41
4.2	Ansatz mit Sample and Hold . . . . .	43
4.2.1	Merkmalsextraktion . . . . .	43
4.2.2	Klassifikation . . . . .	49
<b>5</b>	<b>Ergebnisse und Auswertung</b>	<b>51</b>
5.1	Versuchsaufbau . . . . .	51
5.2	Ergebnisse . . . . .	52
5.2.1	Garnsensor . . . . .	52
5.2.2	Sensor mit silberbeschichteten Stoffen . . . . .	54
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>57</b>
6.1	Zusammenfassung . . . . .	57
6.2	Ausblick . . . . .	57
<b>A</b>	<b>Mikrocontroller</b>	<b>59</b>
A.1	Anleitung zum Aufspielen der Software . . . . .	60
A.2	Pfade zu den Hex- und Eeprom-Dateien: . . . . .	60
<b>B</b>	<b>MIRA Programme</b>	<b>61</b>
B.1	Erklärung FurSensor.xml . . . . .	62
B.2	Erklärung SVMTrainer.xml . . . . .	63
B.3	Anleitung zum Aufnehmen von Daten . . . . .	64
B.4	Anleitung zum Labeln einer <name>.tape Datei . . . . .	65
B.5	Anleitung zum Erstellen einer SVM . . . . .	66
B.6	Besonderheit beim Garnsensor . . . . .	67







# Kapitel 1

## Einleitung

Im Fachgebiet Neuroinformatik und Kognitive Robotik (NIKR) der Technischen Universität Ilmenau wird im Projekt SERROGA an einem Service-Roboter für die Gesundheitsassistenz gearbeitet. Im Rahmen dieser Bachelorarbeit soll eine Erweiterung zu diesem Roboter entwickelt werden. Dem Roboter soll die Möglichkeit gegeben werden, Berührungen wahrzunehmen und diese einordnen zu können. Dazu wird ein Sensor benötigt, der diese Berührungen erfasst. Als Position für ebendiesen wurde die Fläche an der Oberseite des Roboters ausgewählt, wie in Abb. 1.1 zu sehen. Durch die Position sollen die Benutzer dazu animiert werden, typische Berührungen auszuführen, wie sie sie beim Umgang mit (Haus-)Tieren gewohnt sind. Als Berührungen kommen z. B. *Streicheln* oder *Kraulen* in Frage.

Der Sensor selbst besteht aus elektrisch leitfähigen Textilien, damit eine flexible Anpassung an die Oberfläche ermöglicht wird. Des Weiteren ist dieser Sensor von einem blauen Kunstfell überzogen, so dass Assoziation zum Fell eines Tieres hergestellt werden kann. Der Zweck der Wahrnehmung und Erkennung von Berührungen ist es, dem Roboter eine Möglichkeit für die Rückmeldung des Benutzers zu geben, sogenanntes „Feedback“. Durch das Feedback kann der Roboter sein Verhalten bewerten und verändern, z. B. wird eine ausgeführte Aktion nicht wiederholt, wenn diese negativ bewertet wurde oder eine Aktion bevorzugt, wenn der Roboter durch positives Feedback dazu animiert wird. Dieses Lernverhalten zielt darauf ab, den Roboter besser an die Bedürfnisse seiner Benutzer anzupassen. Folgende zwei Eigenschaften sind bei der

---



**Abbildung 1.1:** Blauer Fellsensor auf dem Roboter

*Zu sehen ist die verwendete Roboterplattform, auf der das Fell platziert wurde. Links ist ein mögliches Interaktionsverhalten zu sehen. Rechts ist die Ansicht auf den Roboter von hinten dargestellt. Quelle: [NIKR]*

Erarbeitung zu berücksichtigen: Der Sensor muss zum einen flexibel sein, um sich an die Form des Roboters anpassen zu können und zum anderen möglichst einfach, bzw. sparsam sein. Der zweite Punkt ist insofern wichtig, da der Roboter seine Energie aus Akkumulatoren speist und der Sensor diese nicht übermäßig beanspruchen soll.

In dem nun folgenden Kapitel werden ähnliche Projekte im „State of the Art“ vorgestellt und bewertet. Darauf folgen die Kapitel zu den vom Fachgebiet entwickelten textilen Sensoren, deren Auswertung und dem Ergebnis dieser Arbeit.

---

# Kapitel 2

## State of the Art

In diesem Kapitel findet eine Einführung zum Stand der Technik im Bereich der textilen Sensoren und der Berührungserkennung statt. Dazu wird als Erstes ein allgemeiner Unterabschnitt folgen, in dem zunächst haptische Sensortypen und Einsatzgebiete präsentiert werden. Danach folgen einige spezielle Arbeiten, die sich entweder mit textilen Sensoren oder mit der Erkennung von Berührungen auseinandersetzen.

### 2.1 Allgemeine Einführung

#### 2.1.1 Sensortypen

Für die Erfassung haptischer Signale gibt es eine Vielzahl von verschiedenen Sensortypen. Angefangen bei mechanischen Drucksensoren und Schaltern bis hin zu Sensoren aus leitfähigen Textilien lassen sich diese Sensoren je nach Messgröße oder -Verfahren in vier Kategorien unterteilen: Optik, Kapazität, Schalter und Widerstände [GRIES, 2013].

Bei den optischen Sensoren wird z. B. die Entfernung eines reflektierten Lichtstrahls gemessen. Im Abschnitt 2.3.3 wird ein optischer Sensor beschrieben, der in einem Gehäuse verbaut und senkrecht auf eine reflektierende Fläche gerichtet ist. Wird diese eingedrückt, so verkürzt sich die Entfernung und eine Berührung wird festgestellt.

Bei kapazitiven Sensoren werden Eigenschaften über einen nicht leitenden Stoff be-

---

schrieben. Dieser befindet sich dabei aus zwei elektrisch leitfähigen Stoffen oder Elektroden und verändert die Kapazität z. B. durch Druck.

Die Klasse der Schalter wird durch Kontaktsensoren dargestellt. Wird der Sensor berührt, so schließt sich ein Schalter im System und ermöglicht einen Stromfluss. Dadurch wird ein binärer Wert ausgegeben, der beschreibt, ob der Stromkreis geschlossen ist.

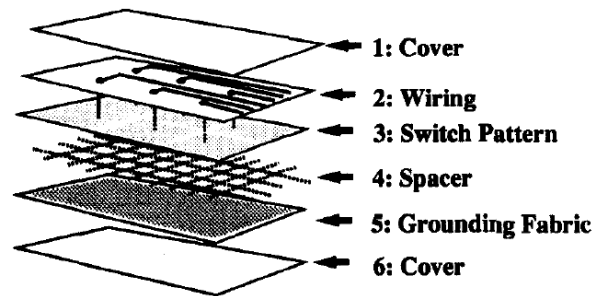
Die vierte Kategorie wird durch Sensoren vertreten, die mit Widerständen arbeiten. Dabei gibt es eine Vielzahl von Konstruktionsvarianten. Im Verlauf dieses Kapitels werden einige Beispiele für diese Verfahren geliefert (Kap. 2.2.2 und 2.2.3). In diesem Zusammenhang sind u. a. piezoresistive Stoffe zu nennen, die bei Druck ihren Widerstand ändern.

### 2.1.2 Einsatzgebiete

Die Einsatzgebiete für taktile Sensoren sind vielfältig. Zum einen spielen sie in der Robotik eine große Rolle, u. a. für die Erkennung von Formen durch Nachbildungen einer menschlichen Hand [BUSCHER et al., 2012], die Kollisionserkennung durch Stoßstangen („Bumper“) oder wie auch in dieser Arbeit, die Erkennung von Berührungen [NAYA et al., 1999]. Zum anderen finden die Sensoren auch in der Medizin wichtige Anwendungsgebiete, beispielsweise als Helfer für minimalinvasive Eingriffe. Dabei unterstützen sie den Arzt, damit dieser mit seinen mechanischen Operationswerkzeugen keine Verletzungen am umliegenden Gewebe verursacht [PUANGMALI et al., 2008] oder durch eine Abtastung einer Gewebefläche eine Diagnose stellen kann [GWILLIAM et al., 2010].

Auch im Alltag sind haptische Sensoren zu finden. Durch das Aufkommen von „Smartphones“ fand eine hohe Verbreitung in Form der „Touchscreens“ statt. Allein für diesen Anwendungszweck gibt es viele verschiedene Varianten für die Umsetzung der Sensoren, z. B. über kapazitive Sensoren [RUAN et al., 2010].

---



**Abbildung 2.1:** Querschnitt des textilen Ganzkörpersensors

*Zu sehen ist der mehrschichtige Aufbau mit zwei Schutzschichten (1 und 6), dem Verkabelungsbereich (2) und den drei Schichten, die den schalterähnlichen Sensor ergeben (3-5). Quelle: [INABA et al., 1996]*

## 2.2 Textile Sensoren

Im folgenden Abschnitt werden nun Sensoren präsentiert, die auf der Basis von leitfähigen Stoffen entwickelt wurden. Damit stellen diese Sensoren eine gute Vorlage und Referenz für die Systeme dar, die vom Fachgebiet NIKR entworfen werden.

### 2.2.1 Ganzkörpersensor

Als Erstes wird das Konzept für einen Ganzkörpersensor präsentiert, welches 1996 entwickelt wurde [INABA et al., 1996]. Besonders hervorzuheben ist an dieser Stelle, dass für die gesamte Fläche des Sensors Textilien zum Einsatz kommen, wodurch eine hohe Anpassungsfähigkeit an den Untergrund erreicht wird. Nun wird der Aufbau des Sensors im Detail gezeigt.

Wie in Abb. 2.1 zu erkennen, besteht der Sensor aus einem mehrschichtigen System. Dabei dienen die beiden äußeren Schichten als Abdeckung zum Schutz des Sensors. Die Schichten drei bis fünf stellen den Sensor dar, der wie ein Schalter funktioniert. Die unterste Schicht (Grounding Fabric) des Schalters besteht dabei aus einem durchgehenden, elektrisch leitfähigen Stoff. Darüber liegt eine nicht leitende Trennschicht (Spacer), die wahlweise aus Baumwolle oder Nylon besteht. Des Weiteren hat diese Schicht die Form eines Gitters, bei dem die Dicke des Stoffs und die Größe der Netz-

löcher einen Einfluss auf die Empfindlichkeit des Sensors haben. Die oberste dieser drei Schichten (Switch Pattern) des Schalters besteht wieder aus elektrisch leitfähigem Stoff. Diesmal ist dieser jedoch in mehrere, voneinander getrennte Segmente unterteilt. In der zweiten Schicht (Wiring) befindet sich die Verbindung des textilen Schalters zum Gerät für die Auswertung der Signale. Dabei besteht die Schicht aus nicht leitfähigem Stoff, in dem leitfähige Fäden verlegt wurden. Dadurch erreicht der Sensor eine hohe Flexibilität und Biegsamkeit, die nicht von elektrischen Bauelementen beschränkt wird. Die Messgröße ist in diesem Fall der Widerstand, der an den einzelnen Segmenten in Schicht 3 gemessen wird. Aus dem Widerstand wird ein binärer Wert errechnet, so dass die einzelnen Sensorflächen entweder den Status *an* oder *aus* haben.

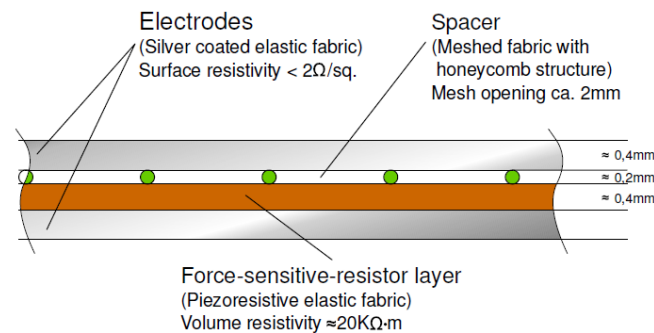
### 2.2.2 Sensorhandschuh

Wie vielversprechend der eben dargestellte, systematische Aufbau ist, zeigt die Arbeit von [BUSCHER et al., 2012]. Im Gegensatz zum Ganzkörpersensor sind die Sensoreinheiten nicht an ein binäres Ergebnis gebunden und können direkt den Wert des Widerstands weiter nutzen. Weiterhin sollte auch erreicht werden, dass sich die Sensoren an komplexere Formen wie z. B. Hände anpassen können. Dies wird u. a. durch die geringe Bauhöhe von 1,5 mm erreicht, durch die auch Finger umhüllt werden können, ohne dass der Sensor zu weit hervorsteht und die Beweglichkeit einschränkt. Des Weiteren kann der Sensor in verschiedenen Größen hergestellt und wie normaler Stoff vernäht werden. Auch bei diesem Sensor handelt es sich um ein mehrschichtiges System.

Bei den äußeren Schichten (Vgl. Abb. 2.2) handelt es sich jeweils um silberbeschichtete, elastische Stoffe, die den Strom leiten. Dazwischen liegen jeweils eine Trennschicht aus nicht leitendem Stoff, welche ein Wabenmuster aufweist, sowie eine druckempfindliche Schicht, die bei Belastung den Widerstand ändert. Die Trennschicht dient dazu, im unbelasteten Zustand einen hohen Widerstand an dieser Stelle zu erzeugen. Ein positiver Nebeneffekt davon ist die Batteriefreundlichkeit, die durch den geringen Stromfluss im unbelasteten Zustand entsteht.

Bei der Herstellung der Sensoren wird ein Ätzprozess genutzt, bei dem bestimmte Formen aus einer großen Sensorfläche heraus geätzt werden, sogenannte „Tacet“. Dabei

---



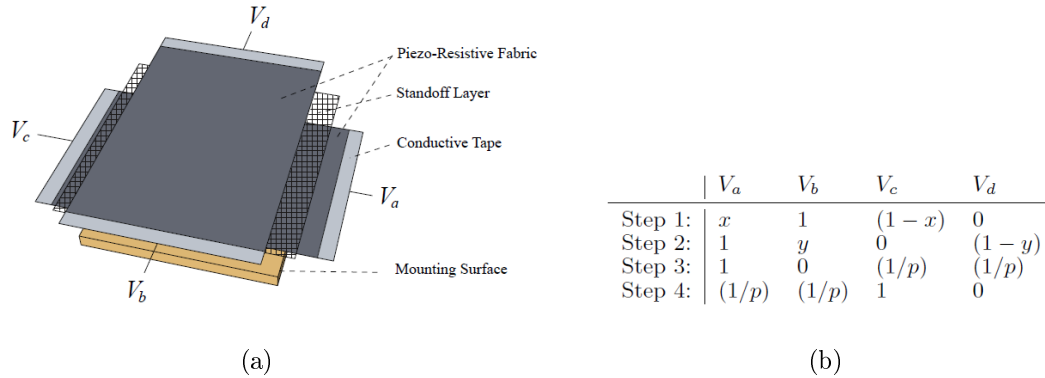
**Abbildung 2.2:** Querschnitt des textilen Sensorhandschuhs

*Zu sehen ist der mehrschichtige Aufbau mit leitfähigen äußeren Schichten (Electrodes), einer druckempfindlichen inneren Schicht (Force-sensitive-resistor layer) und einer Trennschicht aus nicht leitfähigem Stoff, welche in einem Gitter-/Wabenmuster zugeschnitten ist. Quelle: [BUSCHER et al., 2012]*

können dann mehrere Tacel auf einem Stoffstück platziert werden. Durch den Prozess entsteht um die Sensorflächen ein kleiner Rand, der dazu genutzt wird, die Sensoren mit dem gewünschten Objekt zu vernähen. Verläuft die Naht durch den Sensor, so kann das Tacel ein undefiniertes Verhalten zeigen. Jedes Tacel wird durch einen teflonbeschichteten Draht mit dem Empfangsgerät verbunden. Die Kabel verlaufen dabei zum Großteil über den Handrücken. Für den Handschuh wurden 52 Sensoreinheiten verbaut. Jede Einheit ist wegen Störungsvermeidung an einen eigenen Analog-Digital-Umsetzer angeschlossen. Die Messgröße ist auch bei dieser Arbeit der Widerstand, der zwischen den verschiedenen Schichten besteht.

### 2.2.3 Sensorfläche

Bei der textilen Sensorfläche handelt es sich um eine Arbeit von [SCHMEDER und FREED, 2010]. Das Ziel war die Herstellung sehr einfach aufgebauter, textiler Sensoren. Wie in Abb. 2.3(a) zu sehen, besteht der Sensor aus drei wesentlichen Komponenten: Zwei piezoresistive Textilflächen, die über Kreuz gelegt sind, und eine nicht leitende Abstandsfläche dazwischen. Das Ganze wird auf einer entsprechend dimensionierten Auflagefläche montiert. Die vier Kanten des Sensors sind jeweils mit einem Mikrocon-



**Abbildung 2.3:** Aufbau der Sensorfläche und die Beschaltungstabelle

In (a) wird der schematische Aufbau der Fläche gezeigt, dabei werden die drei Komponenten auf einem ebenen Untergrund fixiert. In (b) sind die vier Messschritte abgebildet, aus denen die Werte für die Position  $x$  und  $y$  und den Druck  $p$  abgeleitet werden. Quelle: [SCHMEDER und FREED, 2010]

troller verbunden. Um die Werte für Druck ( $p$ ) und Position ( $x, y$ ) zu erhalten wird mit Hilfe von vier Messschritten Spannung auf die Sensorfläche gegeben, wie in Abb. 2.3(b) abgebildet. Dabei ist je eine Kante auf 1 bzw. 0 gesetzt und die anderen beiden Kanten an einen Analog-Digital-Umsetzer angeschlossen.

Da sich die drei Parameter jedoch nicht linear verhalten, wurde ein Klassifikator für die konkrete Bestimmung eingeführt. Für diesen Zweck wurde eine „Support Vector Machine“ genutzt, wobei die Implementierung der libSVM [CHANG und LIN, 2011] ausgewählt wurde. Um die Klassifikation durchzuführen, wurde die Fläche in  $8 \times 7$  Bereiche mit je fünf Stufen für den Druckwert unterteilt. Des Weiteren wurde auch ein adaptiver, zustandsabhängiger Filter eingeführt, um Rauschen heraus zu filtern, das bei den Messwerten für  $p$  entsteht.

## 2.2.4 Bewertung Sensoren

In diesem Abschnitt folgt kurz eine Gegenüberstellung der vorgestellten Sensoren, dazu dient die Tabelle 2.1. Zu den drei vorgestellten Sensoren wird ein herkömmlicher Drucksensor als Vergleichsmaß genommen.

Der Vorteil des Ganzkörpersensors ist seine hohe Flexibilität, da die Sensorfläche bis



zu den Anschlüssen der Hardware komplett aus leitfähigen Stoffen besteht. Der entscheidende Nachteil ist das binäre Ergebnis. Um Berührungen wie einen *Schlag* oder einen *Klaps* zu unterscheiden, werden genauere Sensorwerte benötigt, da sich diese vor allem im Druck auf den Sensor unterscheiden. Dadurch ist der Sensor eher ungeeignet für das Klassifizieren von druckempfindlichen Berührungsmustern.

Der Sensorhandschuh kann als Weiterentwicklung des Ganzkörpersensors verstanden werden. Bei diesem System ist es möglich, die genauen Messwerte der einzelnen Sensorflächen zu erhalten und weiter zu nutzen. Da hier die Kabel nicht aus Textilien bestehen, muss bei deren Verlegung beachtet werden, dass diese nicht die Beweglichkeit einschränken, wie bei dem vorgestellten Sensorhandschuh oder durch Kabelbruch eine Fehlerquelle darstellen. Dieser Sensortyp ist besonders gut für die zu bewältigende Aufgabe geeignet.

Bei der Sensorfläche, die hier die Aufgabe eines Touchpads hat, ist die Bewertung schwieriger. Aus der Arbeit geht nicht hervor, ob die Sensorfläche immer auf eine ebene Montagefläche angebracht werden muss. Des Weiteren wird nicht deutlich, ob der Aufbau die Flexibilität beeinträchtigt. Im Gegensatz zu den anderen zwei Sensoren muss hier die Position auf der Fläche extra mit Hilfe einer SVM bestimmt werden. Somit ist fraglich, ob mehrere Berührungspunkte gleichzeitig wahrgenommen werden können. Daher ist die Einschätzung der Eignung schwierig und der Sensor wird als ungeeignet eingestuft.

Gegenüberstellung Sensoren				
	textil			herkömmlich
Name	Ganzkörper-sensor (Kap. 2.2.1)	Sensor-handschuh (Kap. 2.2.2)	Sensorfläche (Kap. 2.2.3)	Drucksensor
Vorteile	vollständig aus Textilien (auch Verbindungskabel), flexibel	niedrige Bauhöhe, flexibel, diskrete Messwerte	-	sehr genaue Messwerte, Sensoren voneinander abgeschottet
Neutral	-	herkömmliche Kabelverlegung	Position und Druck werden bereits aus Rohdaten klassifiziert	viele Sensoren
Negativ	nur binäre Werte	jeweils ein ADU für eine Sensoreinheit (Anzahl Sensoreinheiten z. B. 54 Stück)	benötigt Montagefläche, Beugung kann Effektivität verringern	hoher Kabelaufwand, wenig bis gar nicht flexibel, mechanische Bauteile
Eignung	(-) binäre Daten reichen ggf. nicht für Klassifikation aus	(+) textil, flexibel, diskrete Werte	(-) ggf. Probleme bezügl. Flexibilität Zeitverzögerung durch mehrere Klassifikationen	(-) nicht textil, u. U. geringe Flexibilität

**Tabelle 2.1:** In der Tabelle wird die Zusammenfassung der Sensoren mit ihren Eigenschaften aufgeführt. Bewertung: (-) schlecht, (o)neutral, (+)gut

## 2.3 Klassifikation von Berührungen

Auch die Klassifizierung von Berührungsmustern wurde bereits in der Vergangenheit durchgeführt. Auf diese Thematik wird im folgenden Abschnitt eingegangen. Zu erwähnen ist an dieser Stelle, dass bei den Klassifikatoren stets Sensoren zum Einsatz kommen, die nicht aus leitfähigen Textilien bestehen. Daher ist es auch Ziel dieser Arbeit, diese zwei Bereiche zu vereinen.

### 2.3.1 Nächste-Nachbarn-Klassifikation (KNN)

In dem Artikel von [NAYA et al., 1999] wird am konkreten Beispiel die Klassifikation von fünf Berührungsmustern gezeigt. Als Anwendungsfall ist die Gewinnung von Feedback von Benutzern aufgeführt, was ebenfalls dem Ziel dieser Bachelorarbeit entspricht. Bei dem dabei eingesetzten Sensor handelt es sich im Gegensatz zu den vorherigen Sensoren nicht um eine Variante mit leitfähigen Textilien.

Bei diesem Sensor kommt eine Folie (Abb. 2.4(a)) zum Einsatz, die mit druckempfindlicher und leitfähiger Tinte bedruckt ist. Dadurch ist der Sensor sehr dünn und in gewissem Maße flexibel. Die einzelnen gedruckten Sensoren sind so aufgeteilt, dass ein 44 x 44 Elemente Gitter entsteht, wobei die Folie eine Fläche von 16,5 cm x 16,5 cm umfasst.

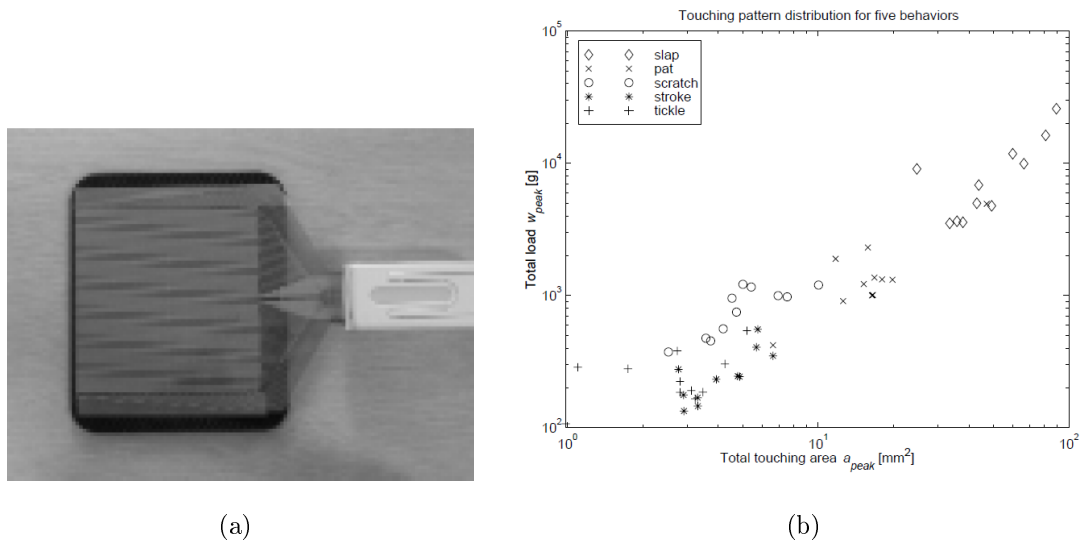
In einem ersten Schritt werden aus den Signalen für jedes Element zwei Merkmale extrahiert. Zuerst wird der Druckwert gemessen und als Merkmal genutzt. Aus diesem Druckwert wird dann mit Hilfe eines Schwellwertes eine binäre Entscheidung getroffen, ob die Fläche berührt wurde.

In einem zweiten Schritt werden diese zwei Merkmale jedes Sensors dann aufsummiert, so dass für jeden Zeitpunkt genau zwei charakteristische Merkmale entstehen, nämlich die Summe aller Sensoren die berührt werden sowie die Summe aller Druckwerte.

Als Berührungsmuster wurden folgende fünf Muster gewählt: *Schlag* (slap), *Hätscheln/Streicheln* (pet), *Streicheln* (stroke), *Kratzen* (scratch) und *Kitzeln* (tickle). Als Klassifikator wird „k-nearest neighbor method with leaving-one-out“ verwendet. Es stellte sich heraus, dass bis auf Streicheln und Kitzeln die Berührungsmuster linear trennbar sind, wie in Abb. 2.4(b) zusehen ist. Um diese zwei übrigen Muster voneinan-

---

der trennen zu können, wurde eine Information über die Zeit eingeführt. Dafür wurden die Werte für Gesamtfläche und -druck zum aktuellen Zeitpunkt  $t$  und zum Zeitpunkt  $t-1$  subtrahiert und mit den Werten vom Zeitpunkt  $t$  normalisiert. Insgesamt konnte eine durchschnittliche Erkennungsleistung von 87,3 % für alle fünf Berührungsmuster erreicht werden.

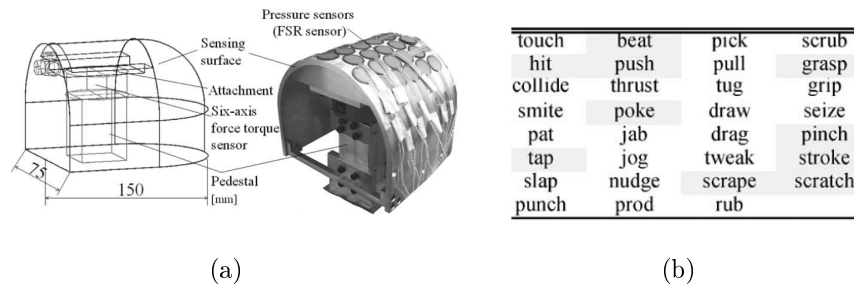


**Abbildung 2.4:** Foliensensor und Grafik für die Verteilung der Muster bei KNN  
Bei (a) ist links die Foliensfläche und rechts die Anschlüsse an die Peripherie zu sehen. Bei (b) sieht man die Verteilung der Berührungen nach Druck und Fläche.  
Quelle: [NAYA et al., 1999]

### 2.3.2 Neuronales Netz (NN)

Im Artikel von [IWATA und SUGANO, 2005] geht es um einen Ansatz, der ein neuronales Netz verwendet. Als Sensoren werden Drucksensoren genutzt, die über den Körper eines Roboters verteilt werden und einige weitere Sensoren im Inneren des Roboters. Zu sehen ist dies in der Abb. 2.5(a), in der der Aufbau und die Verteilung der Sensoren gezeigt wird. Es wurde eine große Anzahl an Klassen definiert, sogenannte PIFACT Terme (unexpected Physical Interference and intended contact).

In der Tabelle der Abb. 2.5(b) sind die 31 Terme aufgelistet, von denen zehn markiert



**Abbildung 2.5:** Versuchsaufbau und PIFACT Terme für das neuronale Netz

In (a) ist der schulterförmige Versuchsaufbau mit den angebrachten Sensoren zu sehen. In (b) ist eine Tabelle zu sehen, in der verschiedene Berührungsklassen zusammengetragen wurden. Die zehn markierten Terme wurden für die tatsächliche Untersuchung ausgewählt. Quelle: [IWATA und SUGANO, 2005]

Force	Max,	Time to reach MAX,	Average Deviation,	Average,	Temporal Gradient,	Spacial Gradient	-
Pressure	Max,	Time to reach MAX,	Average Deviation,	Average,	Temporal Gradient,	Spacial Gradient	-
Contact Area	Max,	Time to reach MAX,	Average Deviation,	Average,	Temporal Gradient,	-	, Summation
High-Pressure Area	Max,	Time to reach MAX,	Average Deviation,	Average,	Temporal Gradient,	-	, Summation
Number of Contact Point	Max,	Time to reach MAX,	Average Deviation,	Average,	Temporal Gradient,	-	-
Two-point Distance	Max,	Time to reach MAX,	Average Deviation,	Average,	Temporal Gradient,	-	-

Moved Distance of Contact Area, Contact Duration, Number of Contact Times

**Abbildung 2.6:** extrahierte Merkmale für das neuronale Netz

Hier ist eine Tabelle der genutzten extrahierten Merkmale zu sehen. Quelle: [IWATA und SUGANO, 2005]

wurden, deren Identifikation mit dem Neuronalen Netz untersucht wird. Auf den Rohdaten der verschiedenen Drucksensoren wird eine Merkmalsextraktion durchgeführt. Dabei werden die in Abb. 2.6 aufgelisteten Merkmale berechnet. Daraus ergibt sich ein großer Inputraum, der von dem Neuronalen Netz verarbeitet werden muss. Bei dem Neuronalen Netz handelt es sich um ein sogenanntes „modified counterpropagation“ Netz.

Der Aufbau des Netzes ist wie folgt: Zuerst kommt die Eingabeschicht, der alle extrahierten Merkmale übergeben werden, darauf folgt eine Schicht für den Wettbewerb der verschiedenen Neuronen und als Ausgabeschicht bekommt jede Berührungsklasse ein

Ausgabeneuron. Dadurch wird nicht nur bestimmt, welche Berührung am wahrscheinlichsten ausgeführt wurde, sondern auch wie hoch die Verwechslungsgefahr mit den anderen Klassen ist. Demnach handelt es sich um eine „one-vs-one“ Klassifikationsstrategie, da am Ende das Neuron ausgewählt wird, welches die höchste Wahrscheinlichkeit hat.

Das für das Verfahren entwickelte System besteht aus einer 500 MHz CPU, die die Sensoren pro Zehntel Millisekunde einmal abtastet, das heißt mit einer Frequenz von 10 kHz. Für die Berechnung der Merkmale und des Neuronalen Netzes wurde eine CPU mit 600 MHz genutzt.

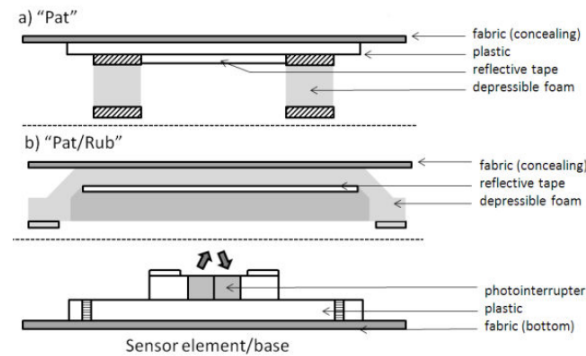
### 2.3.3 Support Vector Machine(SVM)

[COONEY et al., 2012] präsentieren in ihrem Artikel eine Berührungserkennung mit Hilfe von Support Vector Machine (SVM) bzw. KNN. Dabei laufen beiden Varianten unabhängig auf denselben Daten, um die Leistung danach vergleichen zu können. Zu erwähnen ist, dass es in diesem Artikel um eine Fusion von Berührungen und visuellen Informationen handelt, die aber getrennt bewertet werden, so dass in diesem Abschnitt der Fokus auf der Berührungserkennung liegt. Zu beachten ist jedoch, dass auch im Inneren des Roboters Sensoren genutzt wurden, deren Einfluss jedoch unbekannt ist und nicht aus dem Artikel hervorgeht.

Wie in Abb. 2.7 zu sehen, besteht der dafür genutzte Sensor aus einem mehrschichtigen System. Das Grundprinzip ist dabei die Messung des Abstandes zwischen dem Photointerrupter und einem reflektierenden Klebestreifen. Umgeben ist das Innere des Sensors durch verformbaren Schaumstoff. Dadurch sind auch Interaktionen wie Schläge möglich ohne den Sensor zu zerstören. Für den Versuchsaufbau wurden die zwei in der Abbildung beschriebenen Sensoren benutzt. Durch die zwei Varianten sollen zwei Gruppen von Bewegungen unterschieden werden. Zum einen Berührungen, die sich senkrecht auf den Sensor auswirken, wie z. B. ein Schlag und zum anderen parallele Berührungen, wie z. B. Streicheln. Durch eine Reihe von Tests wurden 20 unterschiedliche Berührungsmuster ermittelt.

Als Merkmale wurden Ort, Kraft und Frequenzinformationen extrahiert. Diese Featu-

---



**Abbildung 2.7:** Querschnitt des Sensors mit Photointerrupter für die SVM  
 Abgebildet sind die zwei Sensorvarianten für (a) „Pat“, bzw. (b) „Pat/Rub“. Durch Druck auf die obere Abdeckung (fabric (concealing)) wird der Abstand zwischen der reflektierenden Schicht und dem Photointerrupter verringert. Quelle: [COONEY et al., 2012]

	Touch	Vision	Both
SVMs	71.9%	77.5%	90.5%
k-NN	63.3%	67.4%	82.4%

**Abbildung 2.8:** Vergleich SVM und KNN

Abgebildet sind die Ergebnisse für die SVM und die KNN bzgl. Berührung, visueller Daten und deren Fusion. Relevant ist an dieser Stelle, dass die SVM bessere Ergebnisse geliefert hat. Quelle: [COONEY et al., 2012]

res wurden durch eine Zeitfensterfunktion reduziert und dann an die SVM, bzw. den KNN Algorithmus, übergeben. Dabei wurden aus den Daten für Berührungen insgesamt 320 Merkmale extrahiert. Für die SVM wurde libSVM genutzt [CHANG und LIN, 2011].

Als Ergebnis steht fest, dass in diesem Fall die SVM bei Berührungen besser geeignet ist als der KNN-Algorithmus und dass eine Erkennungsrate von 71,9% erreicht wurde (siehe Abb. 2.8). Dabei wurden 20 Klassen unterschieden. Jedoch kam es auch bei einigen Mustern zu Verwechslungen, die erst durch die Fusion von visuellen Information und Berührungen vermindert werden konnten.

### 2.3.4 Bewertung Klassifikatoren

Wie auch bei den Sensoren folgt an dieser Stelle ein kurzes Resümee über die vorgestellten Klassifikatoren (Vgl. Abb. 2.2).

Der Vorteil des KNN-Verfahrens ist es, dass aus sehr wenig Merkmalen die Klassifikation durchgeführt werden konnte. Das Verfahren stößt an seine Grenzen, wenn viele Klassen getrennt werden sollen, da dann der Rechenaufwand steigt, insbesondere dann wenn sich Klassen sehr ähnlich sind und weitere Merkmale zur Trennung benötigt werden. Das Verfahren eignet sich besonders, wenn wenige, gut trennbare Muster klassifiziert werden sollen.

Bei dem Klassifikator mit Neuronalen Netzen wurde eine aufwendige Merkmalsgewinnung durchgeführt, die in dessen Umfang an Rechenleistung die Kapazitäten dieses Projektes übersteigt. Des Weiteren wird für jede Klasse die Wahrscheinlichkeit ausgegeben, mit der diese Klasse ausgewählt worden wäre. Bei geeigneter Merkmalsgewinnung und Anzahl der Neuronen eignet sich dieser Klassifikator für die zu lösende Aufgabe.

Die SVM konnte eine hohe Anzahl von Klassen mit geringerem Rechenaufwand trennen, als es mit einem KNN-Verfahren möglich war. Da bei der beschriebenen Arbeit jedoch Informationsfusion angewandt wurde, kann daraus die Leistungsfähigkeit der SVM nicht eindeutig bestimmt werden. Allerdings ist auch dieses Verfahren für die Klassifikation von Berührungsmustern geeignet.

---



Gegenüberstellung Klassifikatoren			
Name	k nearest neighbour (Kap. 2.3.1)	Neuronales Netzwerk (Kap. 2.3.2)	Support Vector Machine (Kap. 2.3.3)
Besonderheit	Trennung von fünf Klassen aus vier Features	Trennung von exemplarisch zehn Klassen aus neun Features (mehr Klassen sind möglich)	Trennung von 20 Klassen
Unterscheidungsmerkmal	geringer Aufwand: Messen des Wertes für Druck, Schwellwertentscheidung für binäres Ergebnis, Summation und Subtraktion	Wahrscheinlichkeitswert für jede Klasse (one-vs-one); hoher Aufwand: hohe Abtastrate und rechenintensive Merkmalsextraktion	mittlerer Aufwand: Aufnahme von drei Merkmalen jedes Sensors, Abstraktion dieser Werte durch ein Zeitfenster
Eignung	(+) es werden mit wenig Aufwand fünf Klassen erfolgreich getrennt.	(+) rechenintensiv wegen angestrebter hoher Klassenanzahl und Unterscheidbarkeit.	(+) u. U. werden für ähnliche Klassen weitere Datenquellen benötigt, z. B. visuelle Daten

**Tabelle 2.2:** In der Tabelle wird die Zusammenfassung der Klassifikatoren mit ihren Eigenschaften aufgeführt. Bewertung: (-) schlecht, (o)neutral, (+)gut

## 2.4 Auswertung der Recherche

Aus der Recherche geht hervor, dass die zu bewältigenden Aufgaben zu lösen sind. Es konnte gezeigt werden, dass die Entwicklung der textilen Sensoren weit vorangeschritten ist und ähnliche Ziele wie in dieser Arbeit verfolgt werden. Des Weiteren hat auch die Entwicklung und der Einsatz verschiedener Klassifikatoren zur Berührungserkennung vielversprechende Ergebnisse geliefert.

Das Ziel ist es nun, in Zusammenarbeit mit dem Fachgebiet NIKR die Zusammenführung von textilen Sensoren und Berührungserkennung erfolgreich zu bewältigen. Dabei sind Faktoren wie ein geringer Rechenaufwand und Stromverbrauch ausschlaggebend, um die Energiereserven des mobilen Roboters nicht zu stark zu beanspruchen, sowie die Umsetzung durch Textilien, um sich an die unebenen Flächen des Roboters anzupassen.

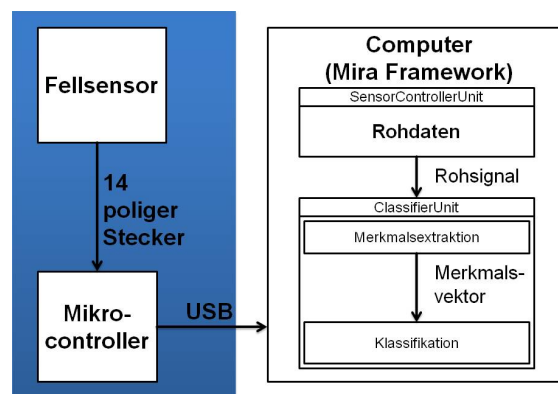
Aus Sicht der Recherche ist es erfolgversprechend, die Prinzipien des Sensorhandschuhs (Kap. 2.2.2) und des KNN-Verfahrens (Kap. 2.3.1) zu kombinieren. Der Aufwand für die Konstruktion ist überschaubar und die Klassifikation einiger weniger Berührungsklassen ist möglich. Das alles geschieht unter der Annahme, dass der textile Sensor eine ähnliche Qualität der Daten erzeugt, wie sie von der bedruckten Folie des KNN-Verfahrens erzeugt wird.

---

## Kapitel 3

# Vom Fachgebiet entwickelte Sensoren

In diesem Kapitel werden die vom Fachgebiet entwickelten textilen, haptischen Sensoren vorgestellt, die auf ihre Eignung zur Berührungserkennung untersucht werden sollen. Dazu wird der jeweilige Sensoraufbau geschildert und die verwendeten Verfahren und Methoden erklärt. Dabei werden die Bereiche Fellsensor und Mikrocontroller in der schematischen Übersicht in der Abb. 3.1 abgedeckt.



**Abbildung 3.1:** Blockbild des Versuchsaufbaus mit Markierung für Sensor

*In der Grafik ist der schematische Aufbau des Versuches zu sehen. Der blau markierte Bereich, der Fellsensor und Mikrocontroller umschließt, wird im folgenden Kapitel behandelt.*

## 3.1 Textiler Sensor mit leitfähigem Garn

### 3.1.1 Technischer Aufbau

Der erste hier vorgestellte Sensorentwurf besteht aus leitfähigem Garn und Kupferkabeln.

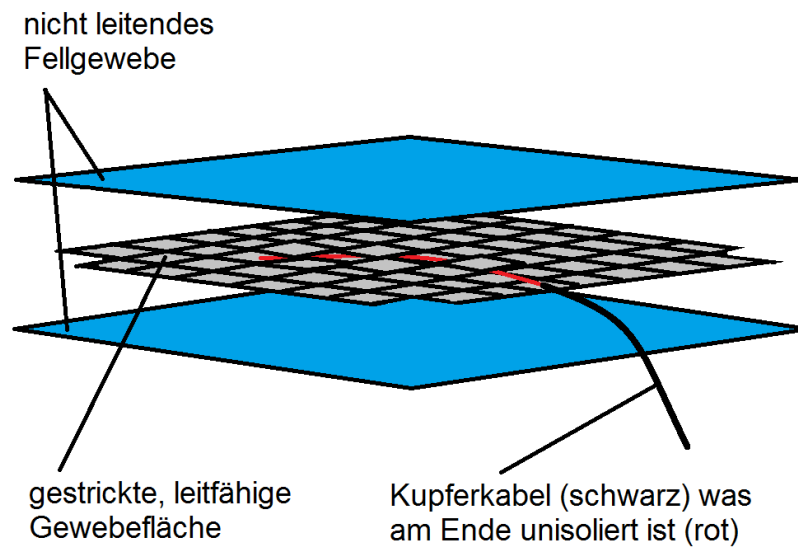
Wie in Bild 3.2 zu sehen, besteht das Fell aus drei Komponenten. Außen befindet sich ein künstliches, blaues und nicht leitendes Fellgewebe. Dieses dient zum Schutz und zum Verdecken der inneren Sensorfläche. Im Inneren befindet sich die zusammenhängende, graue Gewebefläche, die aus leitfähigem Garn zusammengestrickt wurde (siehe Abb. 3.3). In diese Fläche sind Kupferkabel eingelassen, wobei diese an einem Ende unisoliert sind und mit dem anderen Ende am Mikrocontroller angeschlossen werden. Dabei werden zwei unterschiedlichen Anschlüssen am Mikrocontroller unterschieden, denn sechs der vierzehn Anschlüsse haben eine Verbindung zu einem Analog-Digital-Umsetzer (ADU). Durch das unisolierte Ende der Kupferkabel kann nun der Strom durch das Gewebe fließen. Aufgrund des beschriebenen Aufbaus ergibt sich eine einzelne, zusammenhängende Sensorfläche.

Im Bild 3.4 ist die Draufsicht des internen Aufbaus des Fells zu sehen. Als erstes fällt die speziell angefertigte Form auf, die für die ausgewählte Fläche an der Oberseite des Roboters benötigt wurde. Des Weiteren erkennt man die Positionierung der einzelnen Kupferdrähte in der leitfähigen Fläche. Dabei wird nur der Bereich eingezeichnet, an denen die Isolierung der Kupferkabel entfernt wurde. Der nicht sichtbare, isolierte Bereich der Kabel verläuft ebenfalls durch die Fläche und wird am unteren Ende, unterhalb des Kabels mit der Nummer Acht gebündelt und zum Mikrocontroller geführt. Die in der Zeichnung rot markierten Kabel sind dabei jene, welche an die sechs ADU-Einheiten des Mikrocontrollers angeschlossen werden.

### 3.1.2 Messprinzip Variante 1

Das Messen der Werte lässt sich am besten am Ersatzschaltbild erklären, welches in Abb. 3.5 dargestellt ist. Dort sind zwei in Reihe geschaltete Widerstände zusehen, zwischen denen der Anschluss an den ADU liegt. Dadurch wird das Prinzip eines

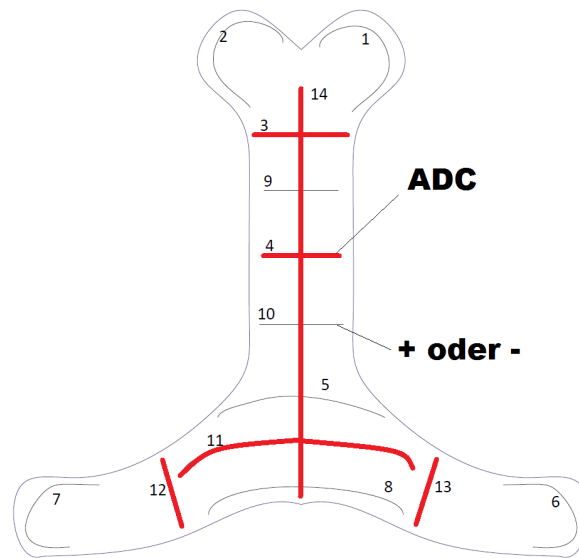
---



**Abbildung 3.2:** Querschnitt des Fellsensors aus gestricktem, leitfähigem Garn  
*Das mehrschichtige System besteht aus dem umhüllenden Fellgewebe, dem zu einer Fläche zusammengestrickten Garn und Kupferkabeln, an deren Enden die Isolierung abgetrennt wurde.*



**Abbildung 3.3:** Foto des leitfähigen Garn  
*Das Bild zeigt die Aufnahme der Sensorfläche aus leitfähigem Garn (grau), welche nach außen gestülpt wurde.*



**Abbildung 3.4:** Draufsicht auf den internen Aufbau des Sensors

*In diesem Bild sieht man die Verteilung der Kupferkabel mit der internen Nummerierung. Die rot markierten Kabel sind jene, welche an die ADU angeschlossen werden sollen. Quelle: [NIKR]*

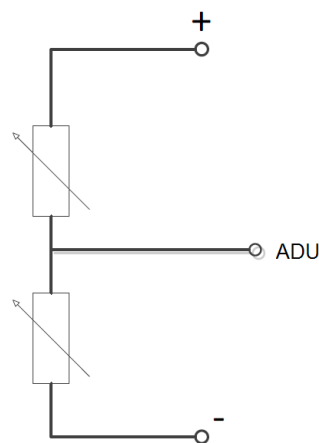
Spannungsteilers umgesetzt. Am Pluspol ist jeweils das stromführende Kabel und am Minuspol jeweils die auf Masse gelegten Kabel angeschlossen. Für dieses Messverfahren wurden 16 Kanäle definiert, die in der Tabelle 3.1 aufgeführt werden. Die Widerstände stellen dabei das leitfähige Gewebe dar. Wird Druck auf das Gewebe ausgeübt, sinkt an dieser Stelle der Widerstand, da die Kontaktfläche zwischen den einzelnen Fasern vergrößert wird, so dass sich auch das Verhältnis zum benachbarten Widerstand ändert. Das beschriebene Verfahren liefert somit die 16 Messwerte des Sensors, aus denen die Berührungen klassifiziert werden sollen.

Als Ergebnis erhält man die Signale, wie sie in Abb. 3.6 abgebildet sind. Zu erkennen ist zum einen, dass jeder Kanal einen anderen Ausgangswert in der Ruhephase hat und zum anderen, dass beim abgebildeten Schlag alle Kanäle beeinflusst werden, obwohl die Berührung eine Ortsabhängigkeit haben sollte. Die Ortsabhängigkeit, bzw. Position sollte aus den längs und quer verlegten Elektroden ermittelt werden.

Der Grund für die auftretenden Probleme liegt in der Konstruktion und den verwend-

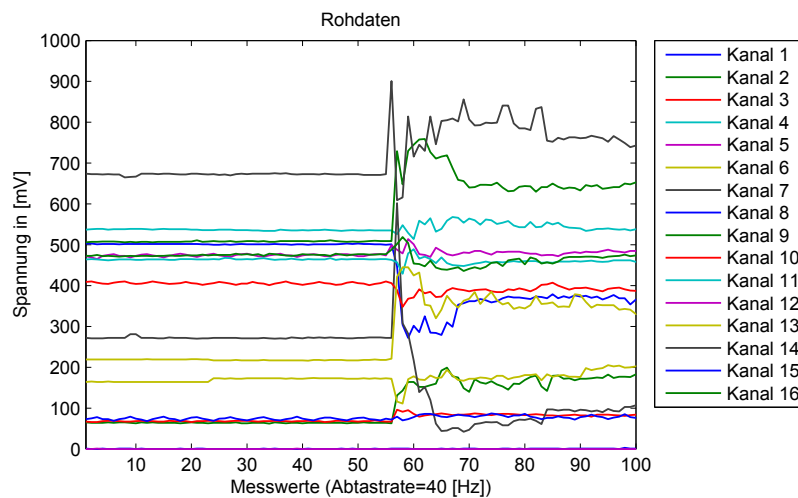
Kabelzuweisung Variante 1							
Kanal	Spg (+)	Masse (-)	ADU	Kanal	Spg (+)	Masse (-)	ADU
1	9	10	4	9	10	9	4
2	1	2,9	3	10	5	7,8	12
3	6	5,8	13	11	8	5,6	13
4	5	6,7,8	11	12	6	5,7,8	11
5	8	5,7	12	13	5	8,9,10	14
6	10	5,8,9	14	14	9	1,2	3
7	2	1,9	3	15	7	5,8	12
8	8	5,6,7	11	16	7	5,6,8	11

**Tabelle 3.1:** In dieser Tabelle ist die Kanalzuteilung für das erste Messprinzip zu sehen. Die Abfrage findet in aufsteigender Kanalnummer statt. Angegeben sind bei Masse und ADU die Kabelnummern, die in Abb. 3.4 vergeben wurden.



**Abbildung 3.5:** Ersatzschaltbild des Messansatzes

Im Bild zu sehen ist das Ersatzschaltbild des Sensors. Dies wird entsprechend den Einträgen aus Tabelle 3.1 beschaltet. Die Widerstände sind dabei das leitfähige Garn, deren jeweiliges Verhältnis gemessen wird. Quelle: [NIKR]



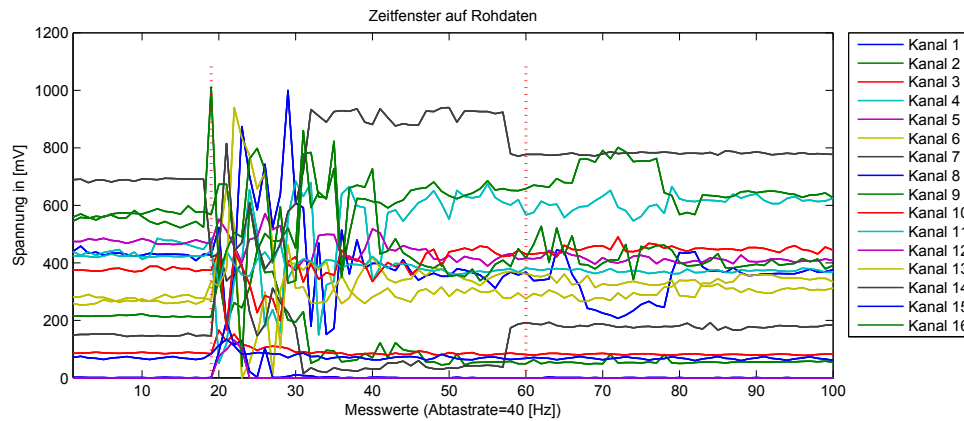
**Abbildung 3.6:** Rohdaten des Fellsensor (gestricktes Garn) V1

*Im Bild zu sehen sind Rohdaten der 16 Kanäle des Sensors. Die unterschiedlichen Spannungswerte ergeben sich aus den unterschiedlichen Entfernungen und Einflüssen der einzelnen Messpunkte. Zu bemerken ist auch das geringe Rauschen. Auf der rechten Seite ist das Muster eines Schlags abgebildet.*

ten Materialien. Für die unterschiedlichen Ausgangszustände sind u. a. die verschiedenen Abstände und leitende Fläche zwischen den einzelnen Elektronen verantwortlich. Da es sich um eine einzelne, zusammenhängende Sensorfläche handelt, entstehen neben den gewünschten Abhängigkeiten auch starke negative Einflüsse wie „Crosstalk“. Wie in der Abbildung zu sehen war, sind fast alle Kanäle beeinflusst worden, jedoch sollte z. B. eine Berührung im Bereich der Elektroden 1, 2 und 3 keinen Einfluss auf den Bereich der Elektroden 5, 7 und 8 haben, da diese physisch voneinander getrennt sind. Dieses Phänomen ist unter der Bezeichnung „Crosstalk“ zu verstehen. Paarweise Abhängigkeiten (z. B. Kanal 1 und 11) sind durchaus für die Positionsbestimmung erwünscht, aber nur, wenn sie physisch im selben Bereich liegen. Alle anderen Einflüsse tragen dazu bei, dass z. B. die Extraktion von Position und Druck erschwert wird.

Durch die Eigenschaften der Fasern im Gewebe ergeben sich weitere Probleme. Eines davon ist in Abb. 3.8 zu erkennen. Ohne äußeren Einfluss verändert sich das Verhältnis von Widerständen, die im selben Bereich der Fläche liegen. Das bedeutet, dass sich das Verhältnis von zwei Flächenstücken verändert hat, die am selben ADU ge-





**Abbildung 3.7:** Einschwingzeit des Fellsensors (gestricktes Garn)

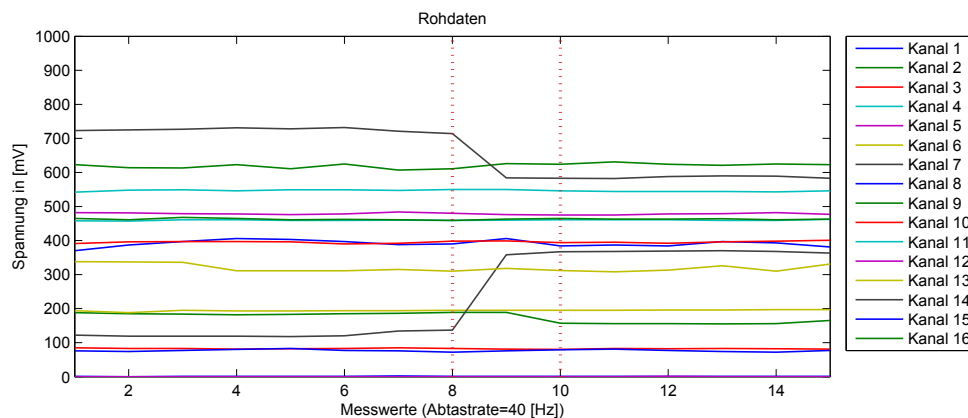
*Die erste Markierung zeigt den ersten Messwert eines Schlags auf das Fell. Die zweite Markierung zeigt in etwa den Punkt, an dem das Fell eingeschwungen ist. Der Unterschied beträgt knapp 40 Messwerte, also etwa eine Sekunde.*

messen werden. Da diese Veränderung des Spannungswertes bereits Einfluss auf das Klassifikationsergebnis nehmen kann, wurde davon abgesehen jeden Kanal einzeln zu betrachten, so dass alle Kanäle gemeinsam verwendet werden.

Des Weiteren hat der Fellsensor durch das Messen von Verhältnissen keinen einheitlichen (Ausgangs-)Zustand. Das wiederum bedeutet, dass der Ausgangszustand willkürlich oder nach einer Interaktion verändert worden sein kann, so dass sich auch die Reaktion des Sensors verändert. War der Wert zum Anfang niedrig und hat sich später auf einem höheren Level eingependelt, so kann bei diesem Kanal aus einem Spannungsanstieg bei einer Interaktion plötzlich ein Spannungsabfall werden.

Das muss im Gegensatz zu den Sensoren aus Kapitel 2 beachtet werden. Dies ist insofern relevant, da die Informationen für eine Berührung sowohl im Spannungsanstieg als auch im Spannungsabfall liegen kann. Unter Umständen kann dann der Anstieg und der Abfall der Spannung jeweils die selbe Berührung bedeuten.

Ein weiteres Problem das sich herausstellte, ist eine lange Einschwingzeit bei den Messwerten. Die Problematik wurde in Abb. 3.7 dargestellt. Zu sehen sind Messwerte, die bei einem kurzen Schlag auf den Sensor gemessen wurden. Dabei beschreibt die erste rote Linie den ersten Messwert der Interaktion. Da die Verweildauer bei einem Schlag



**Abbildung 3.8:** Fehlerquelle in den Rohdaten des Fellsensors (gestricktes Garn)

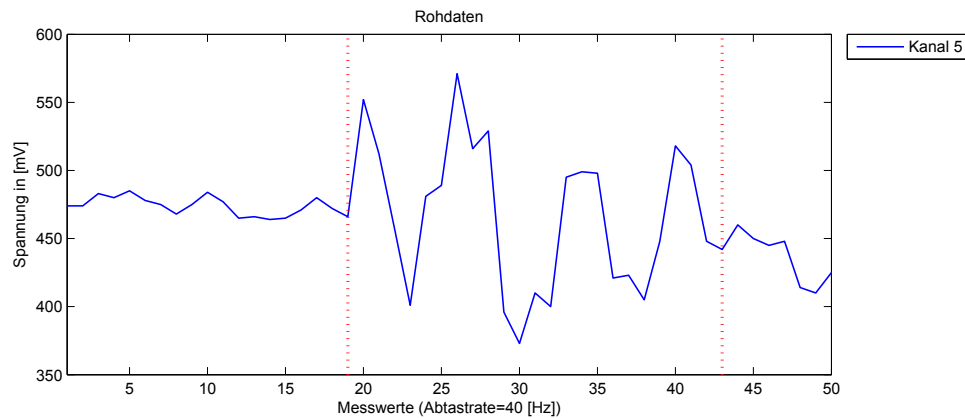
*Zu erkennen ist hier, dass es ohne äußere Einwirkungen zu einer Änderung der gemessenen Verhältnisse kommt.*

nur sehr gering ist, sollte die Hand nach etwa einer Sekunde (Messwert 30) bereits den Sensor wieder verlassen haben. In der Abbildung kann man aber erkennen, dass die Messwerte nicht sofort wieder den Ursprungszustand erreichen und etwa eineinhalb Sekunden (Messwert 60) benötigen, um wieder ein stabilen Wert zu liefern. Das heißt, dass dieser Vorgang eine dreiviertel Sekunde lang andauert. In dieser Zeit ist der Sensor nicht bereit neue qualitative Werte für den Klassifikator zur Verfügung zu stellen. Dies ist darauf zurückzuführen, dass sich die Fasern nach einer Interaktion nur langsam entspannen und dabei unterschiedliche Lagen im Verbund annehmen.

Nach der Beobachtung der genannten Eigenschaften wurde ersichtlich, dass sich die Rohdaten nicht direkt als Merkmale für eine Klassifikation eignen. Um die auftretenden Probleme und Effekte zu minimieren, muss demnach eine Merkmalsextraktion stattfinden.

### 3.1.3 Messprinzip Variante 2

Im Gegensatz zur ersten Variante wird nicht mehr das Verhältnis zwischen zwei variablen Widerständen gemessen, sondern nur noch ein variabler Widerstand gegen einen festen Widerstand, wie im Schaltbild Abb. 3.10 zu sehen. Das wurde dadurch erreicht, dass ein fester Widerstand ( $75\text{ k}\Omega$ ) verlötet, bzw. der interne Pull-up Widerstand (10



**Abbildung 3.9:** Rohdaten eines Kanals des Fellsensors (gestricktes Garn)

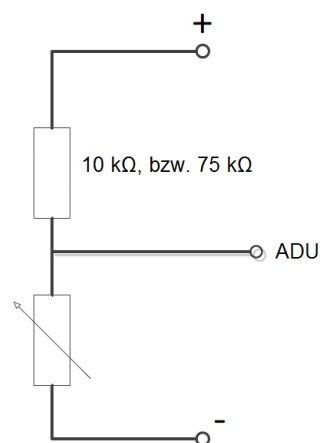
*Bei dem dargestellten Kanal wurde ein Schlag aufgezeichnet. Da der Sensor ständig unter Last steht, können die Informationen sowohl im Spannungsanstieg, wie auch Spannungsabstieg liegen.*

$k\Omega$ ) des ADU Eingangs genutzt wurde. Ansonsten wird das selbe Messprinzip genutzt. Über den festen Widerstand ist das Kabel jeweils an den Pluspol angeschlossen und am Minuspol liegt das Kabel an, zu dem der Widerstand gemessen werden soll. Durch diese Schaltung verursacht ein Druckereignis immer eine Widerstandsverringering, wodurch die gemessene Spannung reduziert wird. Dabei sind die nicht beteiligten Kabel als Eingang geschaltet, so dass großflächige Wechselwirkungen bei der Messung vermieden werden sollen. Wie beim ersten Messprinzip ergibt sich auch bei dieser Variante eine Zuweisung der Kabel zu 16 Kanälen (Vgl. Abb. 3.2). Das Ziel ist es dabei, die Nachteile der ersten Variante zu mindern.

Es gibt zwei Kriterien, in denen dieses Messprinzip besser abschneidet. Zum einen sind die Signalwerte stabiler und haben geringere Schwankungen (Vgl. Abb. 3.11). Zum anderen ist auch die Abhängigkeit zwischen zwei Kanälen verringert worden. Wenn z. B. ein Messstück zwei Kabel umfasst und in zwei Kanälen angeordnet wurden, so hat die Änderung des Verhältnisses auf einem Kanal auch immer eine Änderung auf dem zweiten Kanal verursacht, was jedoch ein unerwünschter Nebeneffekt ist. Bei dieser Variante wird immer nur über ein einziges Kabel pro Kanal gemessen, so dass der Kanal nicht den starken Einflüssen ausgesetzt ist.

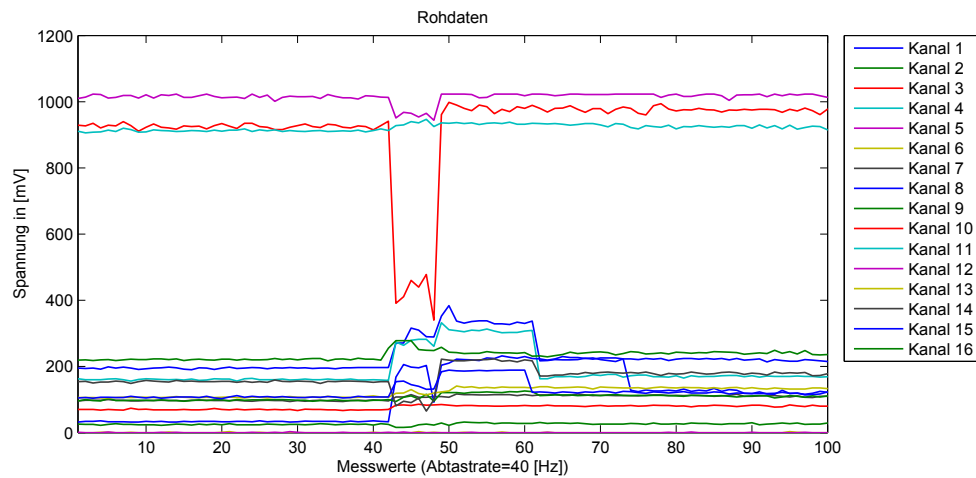
Kabelzuweisung Variante 2					
Kanal	Masse (-)	ADU	Kanal	Masse (-)	ADU
1	1	3	9	9	4
2	5	11	10	2	14
3	7	4	11	8	13
4	1	14	12	12	12
5	6	13	13	9	3
6	8	12	14	10	11
7	2	3	15	10	4
8	8	11	16	5	14

**Tabelle 3.2:** In dieser Tabelle ist die Kanalzuteilung für das zweite Messprinzip zu sehen. Die Abfrage findet in aufsteigender Kanalnummer statt. Angegeben sind bei Masse und ADU die Kabelnummern, die in Abb. 3.4 vergeben wurden.



**Abbildung 3.10:** Ersatzschaltbild des Messansatzes

Im Bild zu sehen ist das Ersatzschaltbild des Sensors. Dieses wird entsprechend den Einträgen aus Tabelle 3.2 beschaltet. Es gibt einen festen Widerstand mit  $10\text{ k}\Omega$  bzw.  $75\text{ k}\Omega$  und einen variablen weiteren Widerstand durch das Fell.



**Abbildung 3.11:** Rohdaten des Fellsensor (gestricktes Garn) V2

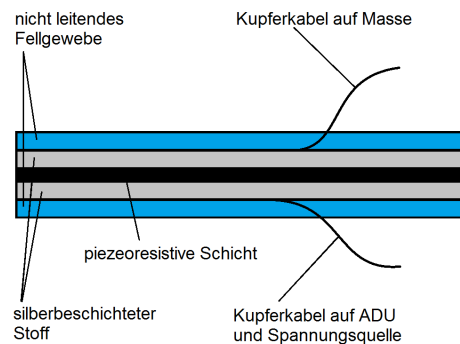
*Im Bild zu sehen sind die Rohdaten der 16 Kanäle des Sensors. Im Vergleich zu Abb. 3.6 sind die Werte stabiler. Hier sind die Auswirkungen zu sehen, die bei einer einfachen kurzen Berührung mit einem Finger geschehen. Gut zu erkennen ist, dass viele Kanäle beeinflusst werden.*

Probleme wie die lange Einschwingzeit nach einer Interaktion oder die Kanäle, die sich immer auf neuen Spannungswerten einschwingen können, wurden dadurch nicht behoben. Deshalb ist auch hier die direkte Nutzung der Rohdaten nicht geeignet.

## 3.2 Textiler Sensor mit silberbeschichteten Stoffen

### 3.2.1 Technischer Aufbau

Aufgrund der Ergebnisse der Rechercharbeiten wurde ein alternativer Entwurf für den Fellsensor erstellt. Dies geschah in starker Anlehnung an den Sensorhandschuh aus Kap. 2.2.2. Dabei wurde jedoch auf eine Trennschicht verzichtet, um die Empfindlichkeit zu erhöhen (Vgl. Abb. 3.12 und Abb. 2.2). Die beiden umhüllenden Schichten werden durch eine piezoresistive Schicht voneinander getrennt, welche ihren Widerstand in Abhängigkeit von Druck verändert. Wird kein Druck ausgeübt, so hat die Schicht einen hohen Widerstand. Dies zeigt sich an den hohen Spannungswerten, die



**Abbildung 3.12:** Querschnitt des Fellsensors aus silberbeschichteten Stoffen

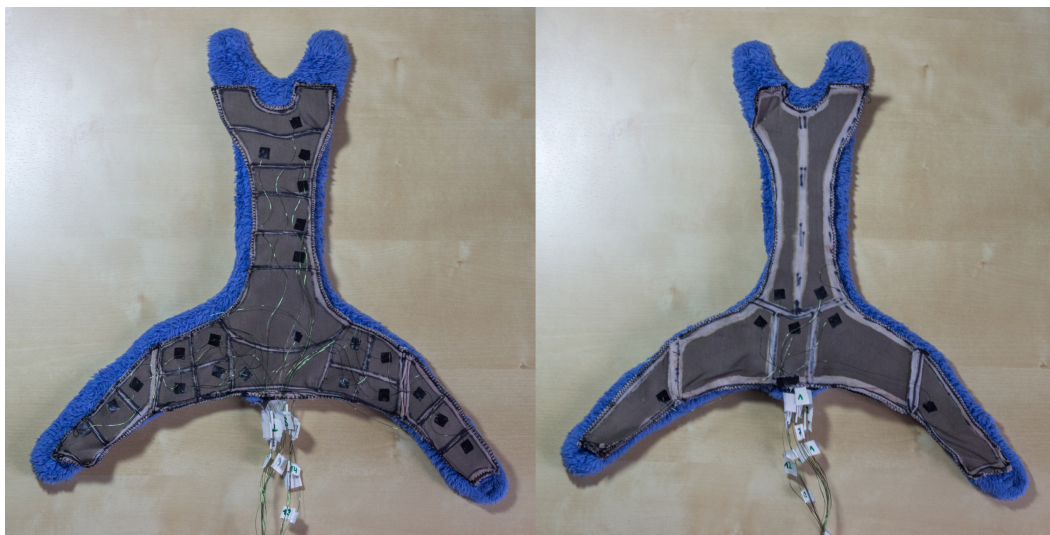
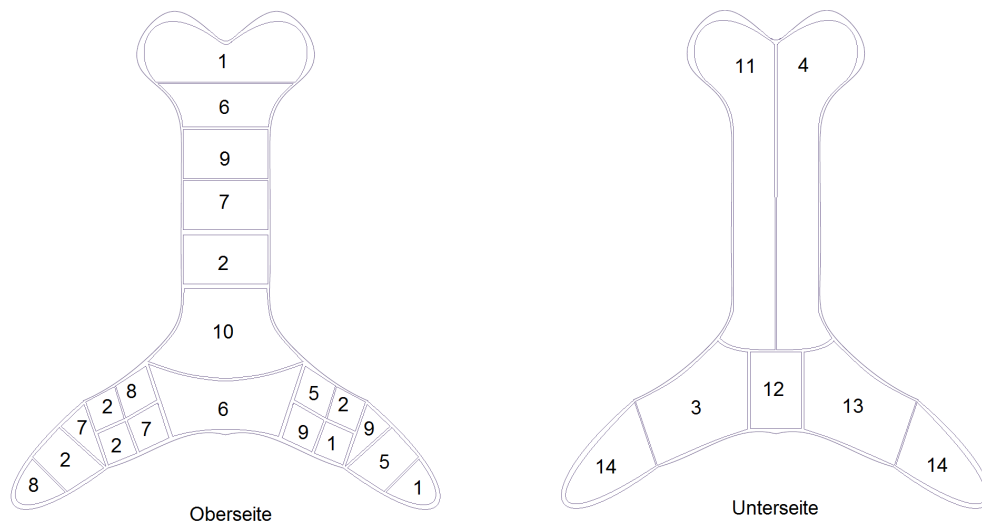
*Das mehrschichtige System besteht aus dem umhüllenden Fellgewebe, den zwei Schichten mit silberbeschichtetem Stoff und den Verbindungskabeln zu Masse, respektive zum ADU, bzw. der Spannungsquelle.*

dort gemessen gemessen werden (siehe Abb. 3.14).

Auch die prinzipielle Herstellung stimmt mit den genannten Sensorhandschuh überein. Mit einer ätzenden Lösung wurden auf einem Stück des silberbeschichteten Stoffs mehrere Flächen ausgeätzt (Vgl. Abb. 3.13). Zu erkennen ist in der Abbildung, dass sich mehrere Flächen der oberen und unteren Schicht überlappen.

Darin besteht ein weiterer Unterschied zum Sensorhandschuh, der einzelne Sensorsegmente hat. Beim Sensorhandschuh hat jedes Sensorsegment einen eigenen Anschluss an die Masse, an den ADU und an die Spannungsquelle. Der in dieser Arbeit verwendete „Atmel ATmega 8“ Mikrocontroller verfügt nur über 14 Anschlusspins, von denen sechs Pins über Anschlüsse an die ADU verfügen. Um mit der Knappheit der Pins umzugehen, wurde eine sequentielle Abfrage der einzelnen Flächen genutzt. Auf der Oberseite sind dazu mehrere Flächen mit der selben Nummer bezeichnet. Das sind diejenigen Flächen, die mit dem selben Pin verbunden werden. Durch den beschriebenen Aufbau wird eine Matrix-Verschaltung realisiert.

Bei der Anzahl der notwendigen ADU wurde ebenfalls der Aufwand reduziert, indem wenige große Flächen verwendet werden, die dann als Gegenstück für mehrere Flächen auf der Oberseite dienen. Dadurch entsteht ein Matrixfeld, bei dem zu einem Zeitpunkt immer nur ein ADU und ein Kabel aktiv sind. Gemessen wird hier die Spannung zwi-



**Abbildung 3.13:** Draufsicht und Foto des Sensors mit silberbeschichteten Stoffen. Zu sehen sind die ausgeätzten Flächen auf den silberbeschichteten Textilien. Links die obere Schicht, deren Flächen alle mit Masse verbunden werden und rechts die untere Schicht, deren Flächen mit ADU und Spannungsquelle verbunden sind. Die Zahlen entsprechen der Belegung der Pins am Mikrocontroller. In den unteren Abbildungen sind Fotos des Versuchsmodells zu sehen. Dabei sind die hellgrauen Flächen jene, bei denen die Silberbeschichtung weggeätzt wurde.

schen den silberbeschichteten Flächen, die sich bei Zusammendrücken der Schichten und des Drucks auf die piezeoresistive Schicht ändert. Alle nicht an der Messung beteiligten Anschlüsse werden hochohmig geschaltet, das heißt, dass der entsprechende Pin als Eingang gesetzt ist.

### 3.2.2 Messprinzip

Das Messprinzip bleibt unverändert zur Variante 2 des Sensors mit leitfähigem Garn im Kaptiel 3.1.3.

Durch den zuvor beschriebenen Aufbau mit überlappenden Flächen entstehen 32 Kanäle (Vgl. Tab. 3.3). Die Verdopplung der Anzahl der Kanäle führt zwangsläufig auch zu einer Reduzierung der Abtastfrequenz, die sich bei stabilen 30 Hz eingependelt hat.

Damit ergeben sich Signale wie in Abb. 3.14. Die Messwerte unterliegen leichten Schwankungen, da auf dem Mikrocontroller kontinuierlich Werte aufgenommen werden und immer das Minimum bis zum Sendezeitpunkt gehalten wird.

Im Vergleich zum ersten Sensorentwurf bietet dieser Sensor ein sehr gutes Ansprechverhalten. Dies zeigt sich insbesondere daran, dass eine sehr kurze Einschwingzeit erreicht wird. Nach einer Interaktion wird sofort ein stabiler Ausgangszustand erreicht. Des Weiteren kann durch das präzise Ansprechverhalten der einzelnen Sensoren eine gute Ortszugehörigkeit festgestellt werden. Die Positionen sind dabei physisch durch den Aufbau gegeben und gut voneinander isoliert. Weiterhin wurden auch keine plötzlichen Spannungsänderung im Ruhezustand festgestellt.

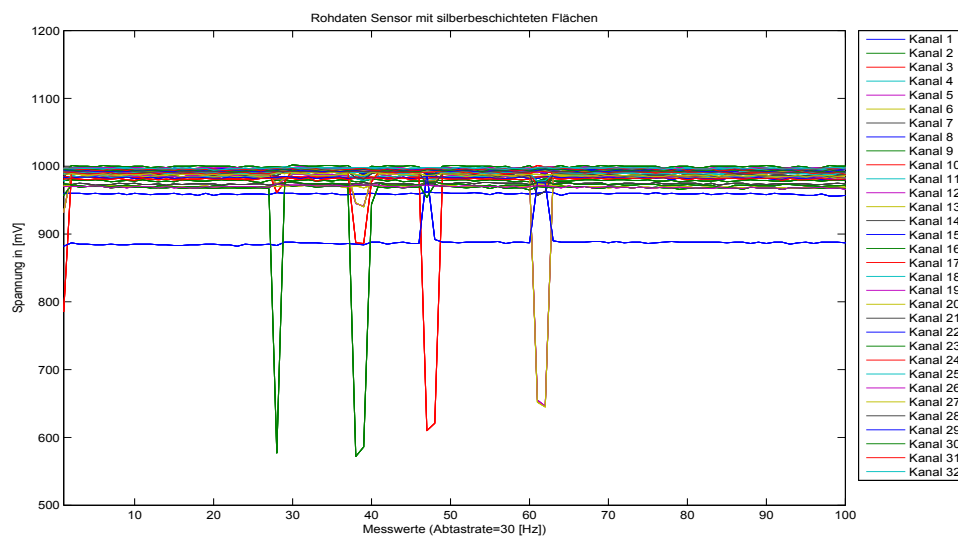
Bei diesem Sensortyp ist es ebenfalls sinnvoll eine Merkmalsextraktion durchzuführen.

---



Kabelzuweisung Drucksensor					
Kanal	Masse (-)	ADU	Kanal	Masse (-)	ADU
1	1	4	17	6	13
2	1	11	18	7	3
3	1	13	19	7	4
4	1	14	20	7	11
5	2	3	21	7	14
6	2	3	22	8	3
7	2	4	23	8	14
8	2	11	24	9	4
9	2	13	25	9	11
10	2	14	26	9	13
11	5	13	27	99	14
12	5	14	28	10	3
13	6	3	29	10	4
14	6	4	30	10	11
15	6	11	31	10	12
16	6	12	32	10	13

**Tabelle 3.3:** In dieser Tabelle ist die Kanalzuteilung für den Sensor mit piezoresistiven Stoffen zu sehen. Die Abfrage findet in aufsteigender Kanalnummer statt. Angegeben sind bei Masse und ADU die Kabelnummern, die in Abb. 3.13 vergeben wurden.



**Abbildung 3.14:** Rohdaten des Fellsensors (silberbeschichtete Stoffe)

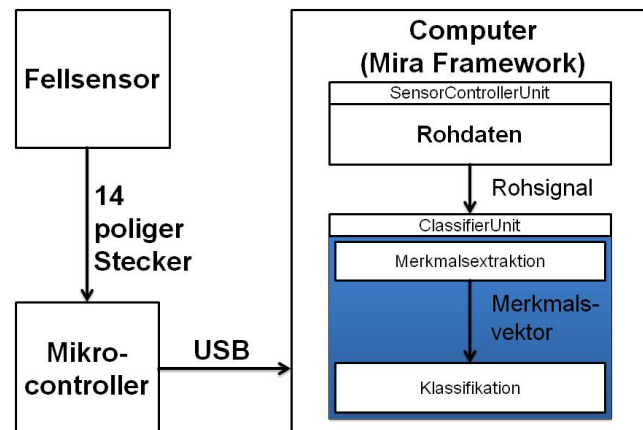
*Im Bild ist der Ruhezustand sowie einzelne Impulse zu sehen, bei denen eine oder mehrere Flächen kurz berührt wurden.*

## Kapitel 4

# Merkmalsextraktion und Klassifikation

Dieses Kapitel beschäftigt sich mit der Merkmalsextraktion und der sich daraus ergebenden Klassifikation. Dies ist notwendig, da sich die Rohdaten alleine nicht für die Klassifikation eignen. Durch eine geschickte Auswahl von Merkmalen wird eine Abstraktion der Daten bewirkt, die z. B. örtlich Einflüsse in eine allgemeine ortsunabhängige Information umwandeln. Aufgrund der Verschiedenheit der Rohdaten der Sensoren ergeben sich auch unterschiedliche Möglichkeiten in der Merkmalsauswahl. Als solche wurde für den Sensor aus leitfähigem Garn ein Zeitfenster-Ansatz und für den Sensor mit silberbeschichteten Stoffen ein „Sample and Hold“-Ansatz gewählt. Damit werden gleichnamigen Bereiche im schematischen Aufbau in Abb. 4.1 erreicht. Dabei werden die Rohdaten vom Mikrocontroller direkt an die Merkmalsextraktion weitergegeben.

---



**Abbildung 4.1:** Blockbild des Versuchsaufbaus mit Markierung für MicroUnit

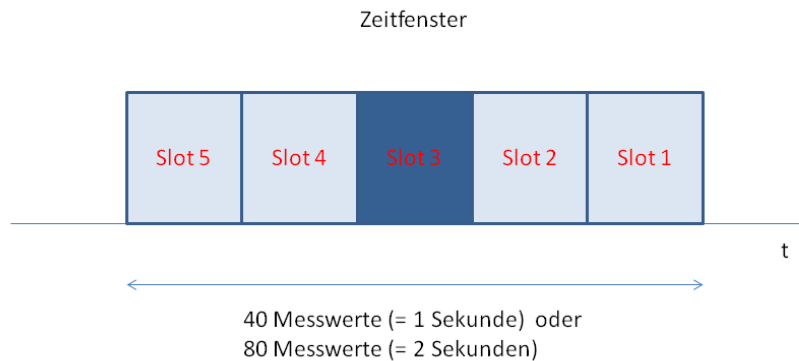
*In der Grafik ist der schematische Aufbau des Versuches zu sehen. Der blau markierte Bereich, der Merkmalsextraktion und Klassifikation umschließt, wird im folgenden Kapitel behandelt.*

## 4.1 Ansatz mit Zeitfenster

### 4.1.1 Merkmalsextraktion

Aufgrund der auftretenden Schwierigkeiten bei den Rohsignalen wurde entschieden, dass eine Extraktion von Merkmalen notwendig ist. Deshalb wurde ein Zeitfenster eingeführt, wie in Abb. 4.2 zu sehen. Dazu müssen die Messwerte aller Kanäle in einem jeweils eigenen Puffer zwischengespeichert werden. Dabei wurden mit den Puffergrößen 40 und 80 gearbeitet, was bei einer Frequenz 40 Hz dann einer Sekunde bzw. zwei Sekunden des Signals entspricht. Die Extraktion der Merkmale findet dann dadurch statt, dass über den Mittelwert des Zeitfensters gearbeitet wird. Dementsprechend wurden die folgenden mittelwertabhängigen Merkmale ausgewählt:

- die Standardabweichung
- die maximale Abweichung vom Mittelwert
- die Anzahl der Vorzeichenwechsel in Relation zum Mittelwert (Nulldurchgänge)



**Abbildung 4.2:** Schema des Zeitfensters

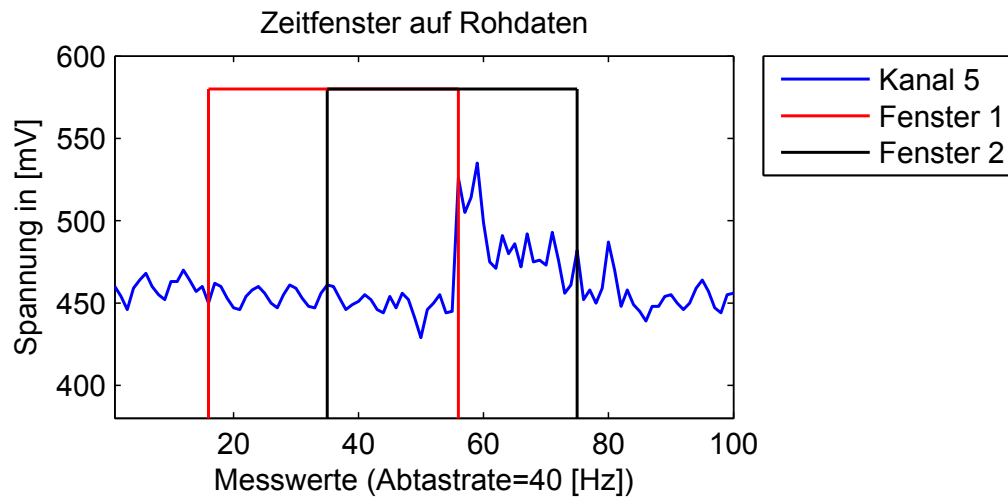
*Zu sehen ist das Zeitfenster, was für die Merkmalsextraktion genutzt wurde. Dies wird in fünf kleinere Slots unterteilt. Über dem Zeitfenster wird der Mittelwert gebildet und daraus in den Slots Merkmale abgeleitet. Der hervorgehobene Slot 3 stellt den Bereich da, der später als positives Trainingsbeispiel gelabelt wird.*

Dadurch sollen die Probleme der unterschiedlichen Ausgangszustände und der Interpretation der Signale verbessert werden. Um die Anzahl der Merkmale zu erhöhen, wurde das Zeitfenster weiterhin in fünf Slots unterteilt. Der Slot mit der Nummer 1 enthält dadurch das jüngste Fünftel der Werte und der Slot mit der Nummer 5 dann das älteste Fünftel der Werte. Die Extraktion geschieht wie folgt:

- Bestimmung des Mittelwertes für das gesamte Zeitfenster
- Berechnung der Standardabweichung, der maximalen Abweichung zum Mittelwert und der Anzahl der Vorzeichenwechsel innerhalb der einzelnen Slots

Dadurch erhält man für die einzelnen Merkmale Werte, die größer als Null sind. Insgesamt ergeben sich aus den 16 Kanälen somit 240 Merkmale.

In den folgenden Abbildungen wird eine Merkmalsextraktion anhand eines Beispiels gezeigt. Begonnen wird mit Abb. 4.3, in der das Originalsignal zu sehen ist. Dabei handelt es sich um einen Schlag auf den Sensor. Weiterhin werden zwei Zeitfenster eingezeichnet, die auch bei den weiteren Abbildungen als Referenzpunkt zu sehen sind. Das rote Zeitfenster umfasst dabei nur einen kleinen Teil am Anfang des Schlagsignals,

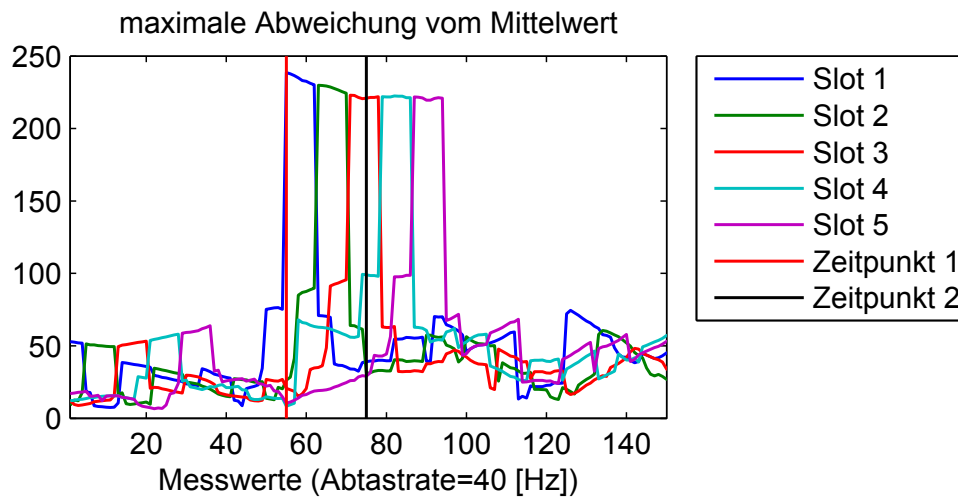


**Abbildung 4.3:** Beispiel Rohdaten mit zwei Zeitfenstern

*Im Bild ist das Muster eines Schlags abgebildet. Über diesem sind zwei Zeitfenster eingezeichnet, die als charakteristische Punkte in Abb. 4.4, 4.5 und 4.6 abgebildet werden. Das schwarze Fenster stellt einen Ausschnitt aus Slot 3 der vorherigen Abbildung dar.*

während das schwarze Fenster das Signal und dessen Veränderung komplett einfängt. Der Abschnitt in dem schwarzen Fenster soll später auch als positives Label in die Klassifikation einfließen.

In der Abb. 4.4 ist die Extraktion der maximalen Abweichung zu sehen. Durch den starken Unterschied der Werte, kann man gut den Unterschied zwischen dem Ruhsignal und dem Signal der Berührung unterscheiden. Am Zeitpunkt 1 kann man gut erkennen, das bereits einige Werte des Signals im Puffer sind und die Kurve einen starken Anstieg verzeichnet. Das Verhalten setzt sich auch entsprechend in den weiteren Slots fort, sobald der Wert in deren Bereich liegt. Die einzelnen Slots haben kurzzeitig einen fast parallelen Lauf zur Achse mit den Messwerten. Dies liegt daran, das der Maximalwert in diesen Slots angekommen ist und sich der Mittelwert nicht verändert, da sich das Signal nach der Interaktion wieder auf einem ähnlichen Wert befindet. Sind die Werte nach einer Interaktion größer oder kleiner, so steigt, bzw. sinkt der Mittelwert und es ist ein leichtes Abfallen/Ansteigen der Kurven über allen Slots zu erkennen. Deshalb entstehen bei der maximalen Abweichung Muster in Form



**Abbildung 4.4:** maximale Abweichung vom Mittelwert

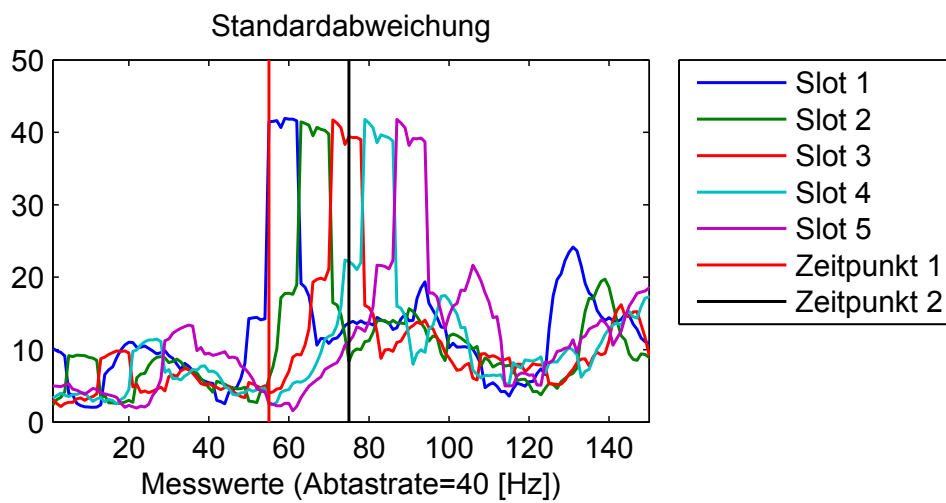
*Aus dem globalen Mittelwert wurde die maximale Abweichung vom Mittelwert innerhalb der Slots extrahiert.*

von Rampen.

Die in Abb. 4.5 zu sehende Standardabweichung weist Ähnlichkeiten im Verhalten zur maximalen Abweichung auf. Jedoch ist die Standardabweichung viel stärker von den Werten im Puffer abhängig, so dass größere Schwankungen zu vermerken sind. Da das Signal vor und nach der Interaktion einen ähnlichen Wert besitzt, ist auch hier zu sehen, wie sich die Muster in den einzelnen Slots wiederholen.

In der letzten Abbildung (Abb. 4.6) zu der Thematik sieht man die Anzahl der Vorzeichenwechsel in Abhängigkeit zum Mittelwert. Dies entspricht den Nulldurchgängen beim Mittelwert. Zu sehen ist, dass die Merkmale vor Zeitpunkt 1 einen hohen Wert haben, da die Werte sehr nahe unter und über dem Mittelwert liegen. Beim Zeitpunkt 2 ist die Anzahl geringer, da die Werte während einer Interaktion stärker in eine Richtung vom Mittelwert abweichen, d. h. das eine Folge von Werten entweder oberhalb oder unterhalb des Mittelwertes liegt.

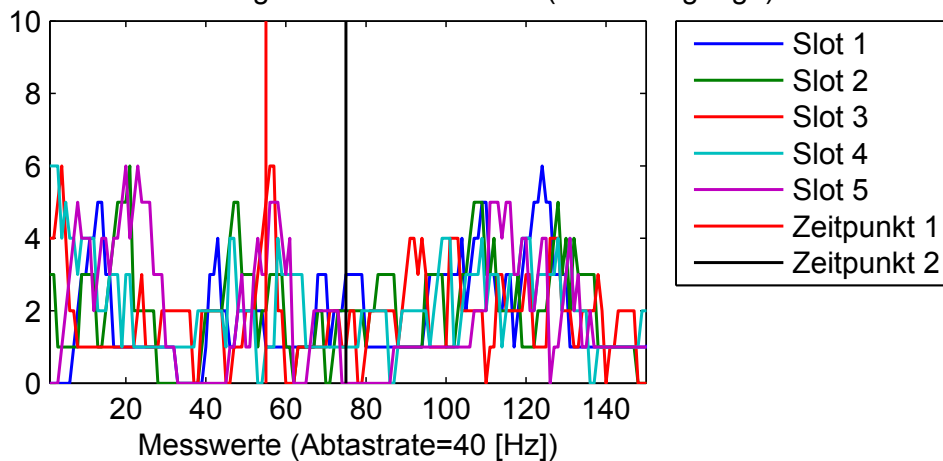
Diese Merkmale stellen die Grundlagen für die weitere Verarbeitung dar. Im darauffolgenden Abschnitt wird nun das Vorgehen für die Klassifikation der Berührungen aus den Merkmalen behandelt.



**Abbildung 4.5:** Standardabweichung

*Aus dem globalen Mittelwert wurde die Standardabweichung innerhalb der fünf Slots ermittelt und dargestellt.*

**Vorzeichenwechsel bezüglich des Mittelwertes (Nulldurchgänge)**



**Abbildung 4.6:** Vorzeichenwechsel vom Mittelwert

*Aus den Informationen der Zeitfenster wurde die Häufigkeit der Vorzeichenwechsel in den Slots im Vergleich zum Mittelwert extrahiert.*



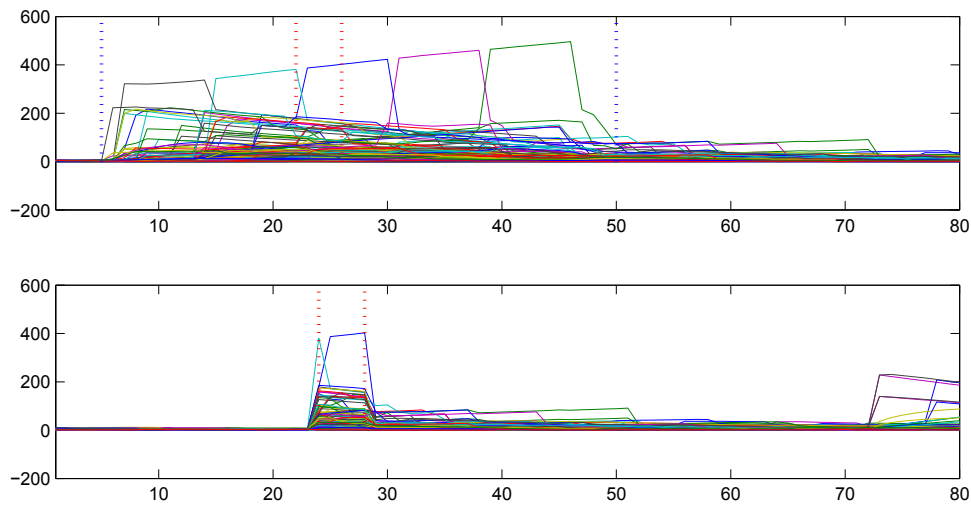
### 4.1.2 Klassifikation

Dieser Abschnitt behandelt die Vorgehen, die zur Klassifikation unternommen wurden. Als erstes wurden vier Berührungsmuster festgelegt, die klassifiziert werden sollen. Ausgesucht wurden folgende Muster: *Schlag*, *Streicheln*, *Kraulen* und *Klaps*. Der *Klaps* stellt dabei eine Art sanften *Schlag* dar, der eine positive, bestätigende Assoziierung darstellt. Die Wahl und die Anzahl dieser Berührungen begründet sich aus den Rechercharbeiten, bei denen auch diese Muster häufig vorkamen. Da die Anzahl so klein ist, liegt daran, dass der Sensor und die Klassifikation Thema dieser Arbeit sind. Deshalb wurden vier Berührungen ausgewählt, die verschiedene Berührungsmuster erzeugen. Sofern diese korrekt klassifiziert werden, kann auch die Anzahl der Berührungsklassen erhöht werden, um die Leistungsfähigkeit des Sensors zu prüfen und damit mehr Interaktionsmöglichkeiten zur Verfügung zu stellen.

Als zweites wird auf die Auswahl des Klassifikators eingegangen. Wie die Recherche in Kap. 2 zeigte, eignen sich die dort vorgestellten Klassifikatoren gleichermaßen für das zu lösende Problem. Der KNN-Algorithmus wurde in der Recherche als vielversprechend angesehen, jedoch zeigte eine der anderen Arbeiten, dass die SVM bessere Ergebnisse liefert. Deshalb wurde die SVM ausgewählt, die Wahl fiel hierbei auf die Implementierung in der libSVM [CHANG und LIN, 2011]. Diese wurde mit den Standardparametern genutzt. Das heißt, der Typ der SVM ist „C\_SVC“ und die Kernel Funktion ist die „Radial Basis Function“ (RBF).

Weiterhin wird das Verfahren im „one-vs-all“-Prinzip genutzt, so dass jede Klasse mit allen Trainingsdaten trainiert wird und nur die gewünschte Klasse daraus klassifiziert werden soll. Dadurch wurde es notwendig, dass für jede Berührung eine eigene SVM genutzt wurde. Weiterhin ergibt sich dadurch ein Problem bezüglich der finalen Entscheidung. Im Moment wird von jeder SVM ein Ergebnis ausgegeben, so dass mehrere SVMs zur selben Zeit eine positive Klassifizierung berechnet haben können. Zur Lösung gibt es verschiedene Möglichkeiten. Einerseits können die Ergebnisse durch einen weiteren Klassifikator auf ein Ergebnis reduziert werden. Dabei müssten die jeweiligen SVMs Wahrscheinlichkeitswerte ausgeben. Daraufhin kann z. B. mittels Schwellwerte und fester Rangfolge der Priorität gearbeitet werden oder es wird ein Klassifikator

---



**Abbildung 4.7:** gelabelte Daten mit und ohne Pufferbereich

*Im oberen Bild ist das Originalsignal zu sehen, im unteren das beschnittene Signal.*

*Der positiv gelabelte Bereich ist zwischen den roten Linien. Im oberen Bild wird das Stück von der blauen zur roten bzw. von der roten zur blauen Linien ausgeschnitten.*

mit mehreren Klassen genutzt. Andererseits könnte auf direkt eine Mehrklassen SVM umgestellt werden, die dann im „one-vs-one“ aus mehreren Klassen nur ein Ergebnis zurückgibt.

Die für die Klassifikation erforderlichen Trainingsdaten wurden folgendermaßen erstellt. Für eine positive Bewertung einer Klasse wurde immer der dritte Slot aus dem vorgestellten Zeitfenster in Abb. 4.2 genutzt. Nämlich genau dann, wenn die Interaktion diesen Slot erreicht hat. Das bedeutet, dass die Interaktion in der Mitte des Zeitfenster angekommen ist und sowohl Informationen über den vergangenen Verlauf als auch den weiteren Verlauf der Interaktion vorhanden sind. Dies ist u. a. notwendig, damit ein impulsartiges Muster, wie beispielsweise ein *Schlag*, von einem andauernden Muster, wie ein *Streicheln*, unterschieden werden kann. Ein Nachteil dabei ist, dass das Ansprechverhalten sich verschlechtert, da nach einer Interaktion ca. eine halbe Sekunde verstreicht, bis eine Entscheidung getroffen wird.

Um die Klassifikationsergebnisse zu verbessern, wurde der Pufferbereich um den po-

sitiv markierten Bereich aus den Daten gelöscht (Vgl. Abb. 4.7). Dadurch werden Nebeneffekte wie „Overfitting“ reduziert.

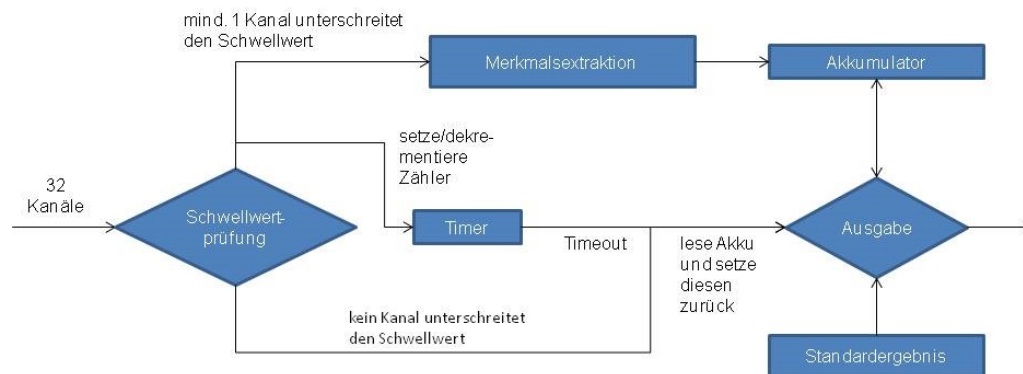
## 4.2 Ansatz mit Sample and Hold

### 4.2.1 Merkmalsextraktion

Die Merkmalsextraktion beim zweiten Sensor gestaltet sich wesentlich einfacher als beim ersten Sensor aus Garn. Das liegt daran, dass durch das gute Ansprechverhalten und die geringen Störungen ein guter Positions- und Druckwert gegeben ist. Die Kanäle sind alle mit einer festen Position verbunden, dadurch lässt sich die genannte Ortszugehörigkeit ermitteln.

Um die weiter unten genannten Merkmale zu extrahieren, wird ein „Sample and Hold“-Ansatz benötigt. Dies dient vor allen Dingen der Gewinnung von Informationen über die Zeit, die dadurch in die Merkmale einfließt. Der Ansatz funktioniert wie in Abb. 4.8 dargestellt. Als erstes wird überprüft, ob einer der Kanäle den Schwellwert unterschreitet. Ist dies der Fall, so wird die Hold-Phase begonnen und die Merkmale werden extrahiert. Die Hold-Phase hält solange an, wie die Schwellwertbedingung nacheinander erfüllt wird. Des Weiteren wird auch ein Timer gestartet, der bei jeder Iteration der Hold-Phase herunter gezählt wird. In jeder Iteration werden die Merkmale neu berechnet und entsprechend im Akkumulator abgelegt. Wird die Schwellwertbedingung nicht erfüllt, d. h. kein Kanal unterschreitet den Schwellwert, so wird der Akkumulator einmal ausgelesen und danach auf den Ausgangswert zurückgesetzt. Das gleiche geschieht, wenn der Timer abgelaufen ist. Während der Hold-Phase wird nur der Standardwert zurückgegeben, was in diesem Fall eine Folge von vier Nullen darstellt.

---



**Abbildung 4.8:** Blockbild des Sample-and-Hold-Ansatzes

*Zu sehen ist das Vorgehensmodell des Sample-and-Hold-Ansatzes. Solange der Schwellwert unterschritten wird, werden Merkmale extrahiert und akkumuliert (sampling). Wird von keinem Kanal der Schwellwert unterschritten oder tritt ein Timeout auf, so wird der Akku einmal ausgelesen und zurückgesetzt (holding). In allen anderen Fällen wird ein Nullvektor als Standardergebnis ausgegeben.*

Aufgrund der Ausgangslage und dem vorgestellten Verfahren wurden folgende vier Merkmale extrahiert:

- Die Anzahl aktivierter Flächen  
Unter Nutzung eines Schwellwertes wird entschieden, ob eine Fläche aktiviert ist oder nicht. Es wird jeweils das Maximum gespeichert.
- Der maximale Druck  
Ergibt sich aus der Summe aller Messwerte der aktivierten Flächen. Es wird ebenfalls das Maximum gespeichert.
- Die Länge des Pfades  
Aus dem gewichteten Mittelpunkt der aktivierten Flächen wird zu jedem Zeitschritt ein Pfad berechnet und gespeichert.
- Die maximale Distanz zum ersten Punkt  
Die Distanz vom ersten gewichteten Mittelpunkt zum weitesten entfernten gewichteten Mittelpunkt wird gespeichert.

Durch die Wahl dieser Merkmale werden die Ortsinformationen abstrahiert.

Die maximale Distanz zum ersten Punkt spiegelt den Zustand der Bewegung wieder. Die stationäre Berührung hat nur einen geringen Wert, da sich der gewichtete Mittelwert nicht sehr weit bewegt. Dies hat einen Einfluss auf die Klassifikation, da dadurch die eher stationären Klassen *Schlag*, *Klaps* und *Kraulen* von der Bewegung *Streicheln* unterschieden werden können (siehe Abb. 4.9).

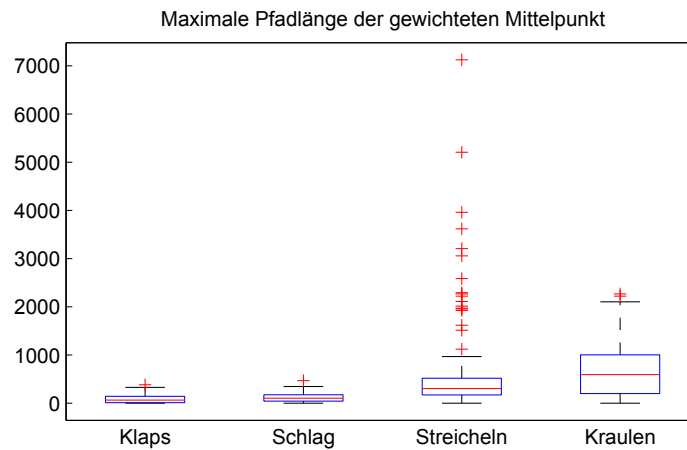
Die Länge des Pfades hingegen spiegelt den zeitlichen Verlauf wieder. Je länger der Pfad wird, desto mehr Hold-Phasen wurden durchlaufen und desto mehr Zeit ist verstrichen. Dies trägt dazu bei, die impulsartigen Klassen *Schlag* bzw. *Klaps* von den zeitlich längeren Klassen *Streicheln* und *Kraulen* zu unterscheiden (siehe Abb. 4.10). Beide Merkmale tragen in Verbindung dann dazu bei, das *Streicheln* und *Kraulen* voneinander zu trennen.

Für die Unterscheidung von *Schlag* und *Klaps* ist insbesondere die Summe der Druckwerte von Bedeutung, da dies das Hauptunterscheidungsmerkmal darstellt (siehe Abb. 4.11).

Die Anzahl der aktivierten Flächen wurde zusätzlich noch eingefügt und hat einen Einfluss auf die Klassifikation bei allen Klassen (siehe Abb. 4.12). Wenn *Schlag* und *Klaps* voneinander getrennt werden sollen, kann dies insofern von Bedeutung sein, da beim impulsartigen *Schlag* mehr Kraft aufgewendet wird, wodurch mehr Flächen den Schwellwert unterschreiten können. Beim *Streicheln* wird so z. B. die ganze Handfläche aufgelegt, aber nicht alle berührten Flächen werden aktiviert. In der Abb. 4.13 sind sowohl die Rohdaten als auch die gewichteten Mittelpunkte eines Streichelmusters dargestellt. Aus den gewichteten Mittelpunkten (rote Vierecke) wird dann die Pfadlänge sowie die maximale Entfernung zwischen Startpunkt und allen weiteren Punkten berechnet.

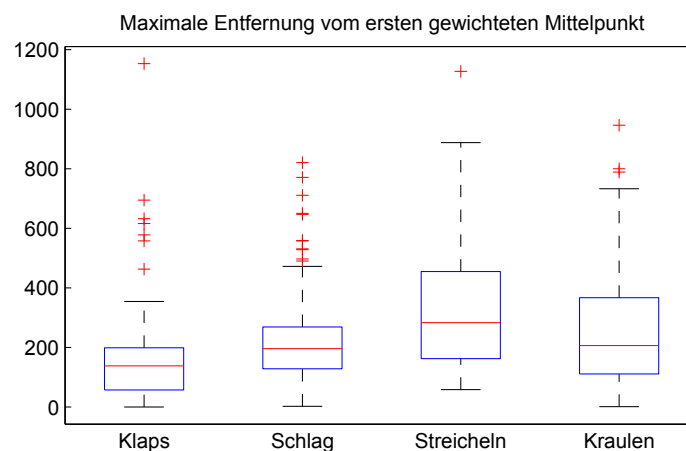
Durch diese Variante des Fellsensors und des Vorgehens konnte die Anzahl der notwendigen Merkmale von 240 beim ersten Sensorentwurf auf vier reduziert werden. Des Weiteren wurde im State of the Art (Kap. 2) gezeigt, dass einige dieser Merkmale sehr aussagekräftig für die Klassifikation von Berührungen sind.

---



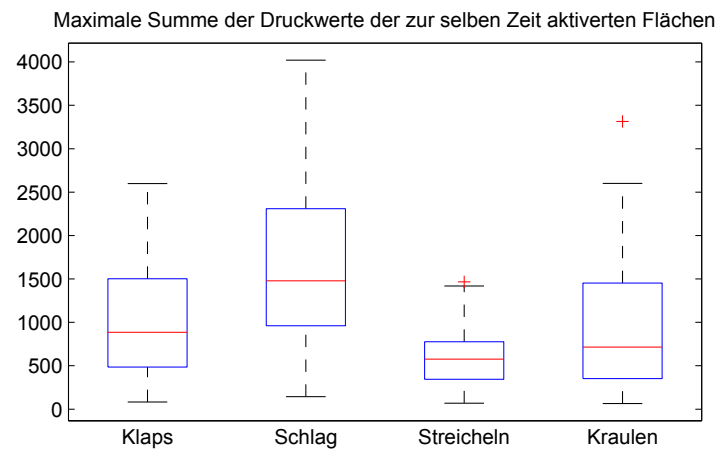
**Abbildung 4.9:** Pfadlängen aller vier Klassen nebeneinander

*Zu sehen sind die vier Klassen und ihre aufgenommenen positiven Trainingsmuster. Gut zu erkennen ist die Unterscheidbarkeit von Streicheln/Kraulen und Schlag/Klaps.*



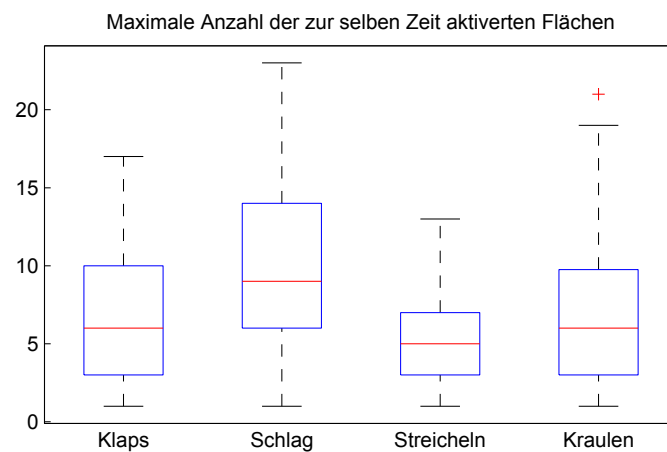
**Abbildung 4.10:** Maximale Abweichung vom ersten gewichteten Mittelpunkt

*Hier sind wieder die vier Klassen nebeneinander dargestellt. Im Durchschnitt wird beim Streicheln die größte Entfernung erreicht. Durch ungleichmäßiges Aufsetzen bei den Berührungen kann aber auch bei den anderen Klassen eine hohe Entfernung gemessen werden.*



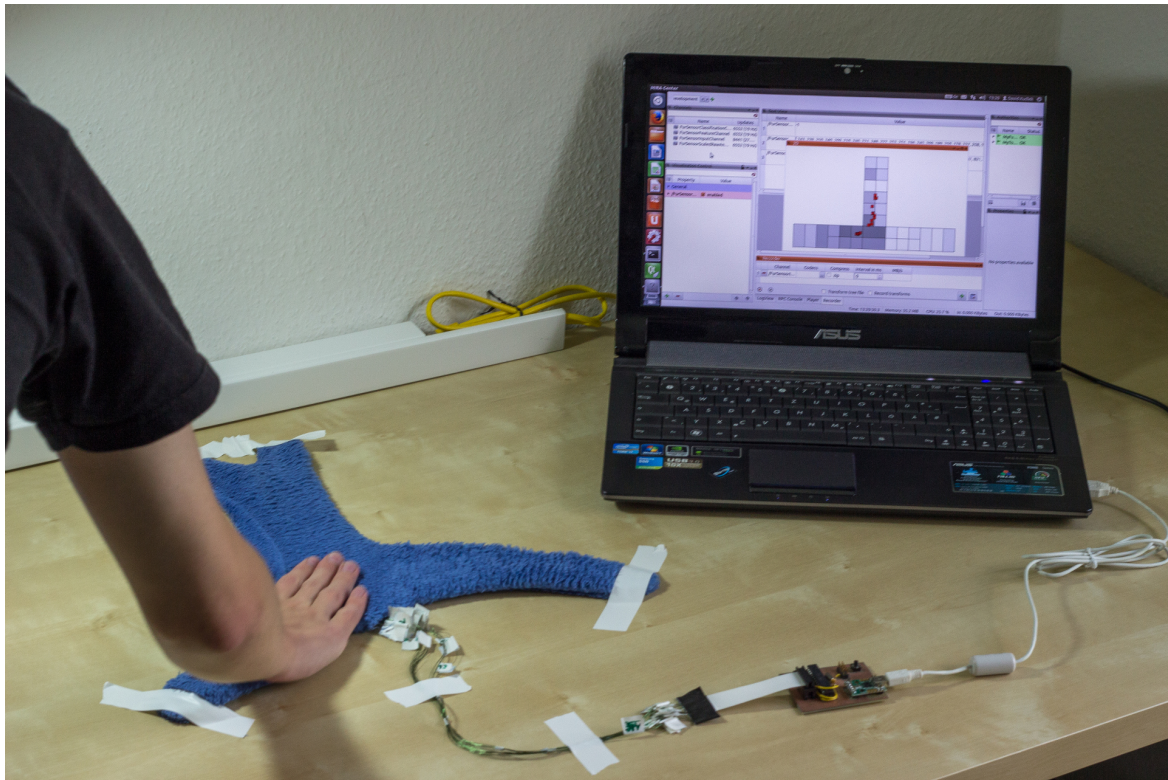
**Abbildung 4.11:** Maximale Summe der Druckwerte der aktivierten Flächen zu einem Zeitpunkt

*Zu erkennen ist hier, dass Schlag und Klaps höhere Werte erreichen. Dies liegt daran, dass bei den Berührungen zumeist die ganze Handfläche aufliegt, wodurch viele Flächen aktiviert werden und dies z. B. beim Schlag auch mit erhöhtem Druck verbunden ist.*



**Abbildung 4.12:** Die größte Anzahl an aktivierten Flächen zu einem Zeitpunkt

*Hier wird wieder derselbe Trend wie in Abb. 4.11 sichtbar. Durch den Bewegungscharakter bei Streicheln und Kraulen wird die Handfläche nur zum Teil aufgelegt oder nur mit geringem Druck, so dass weniger Flächen aktiviert werden.*



**Abbildung 4.13:** Foto einer Streichelbewegung

*Auf dem Monitor auf der rechten Seite ist eine Visualisierung zu sehen. Dort erkennt man die Abbildung eines Streichelmusters. Dabei ist der momentane Druck auf die verschiedenen Flächen zu sehen (graue Abstufungen), sowie eine Menge von roten Vierecken, die die gewichteten Mittelpunkte der letzten aufgenommenen Zeitschritte darstellen. Die Bewegung erfolgt von oben in der Mitte nach unten links.*



### 4.2.2 Klassifikation

Es wurden sowohl die vier Klassen als auch die SVM als Klassifikator beibehalten, so wie sie bereits im Kap. 4.1.2 erwähnt wurden. Für die Klassifikation selbst mussten keine weiteren Schritte unternommen werden, da jede Berührung nur zu einem einzigen Zeitpunkt durch eine Ausgabe wahrgenommen wird. Wenn also die Trainingsdaten für einen *Schlag* aufgenommen wurden, so bedeutet jede Abweichung eines Merkmals von Null eine positives Label. Daraus folgt weiterhin, dass sich das Labeln der Trainingsdaten automatisieren lässt und nicht aufwendig per Hand durchgeführt werden muss.

Die Merkmale lassen sich jedoch nicht vollständig eindeutig den Klassen zuordnen (Vgl. Abb. 4.9, 4.10, 4.11 und 4.12). Daher wurde statt der 1-vs-all-SVM für jede Berührung auf eine 1-vs-1-SVM mit mehreren Klassen umgestellt. Statt für jede Berührung eine eigene 1-vs-all-SVM zu trainieren, wird eine einzige SVM trainiert, die für jede Berührung ein eigenes Label hat. Dies sorgt dafür, dass die SVM immer eine Klasse ausgibt, wenn eine Berührung erkannt wurde. Bei der ersten Variante trat häufiger der Fall auf, dass keine Klasse erkannt wurde. Aufgrund der Ähnlichkeit der Merkmale mancher Klassen ließ sich dies nicht vermeiden. Mehrere SVMs zu nutzen führte auch zu anderen negativen Aspekten, zum einem hat jede weitere SVM einen Leistungsverlust verursacht, zum anderen konnten zur selben Zeit mehrere Klassen erkannt werden, wodurch eine weitere Filterung notwendig geworden wäre. So trägt die Mehrklassen-SVM auch dazu bei, dass die Skalierbarkeit beim Hinzufügen weiterer Klassen erhalten bleibt und immer nur ein Ergebnis ausgegeben wird.

---



## Kapitel 5

# Ergebnisse und Auswertung

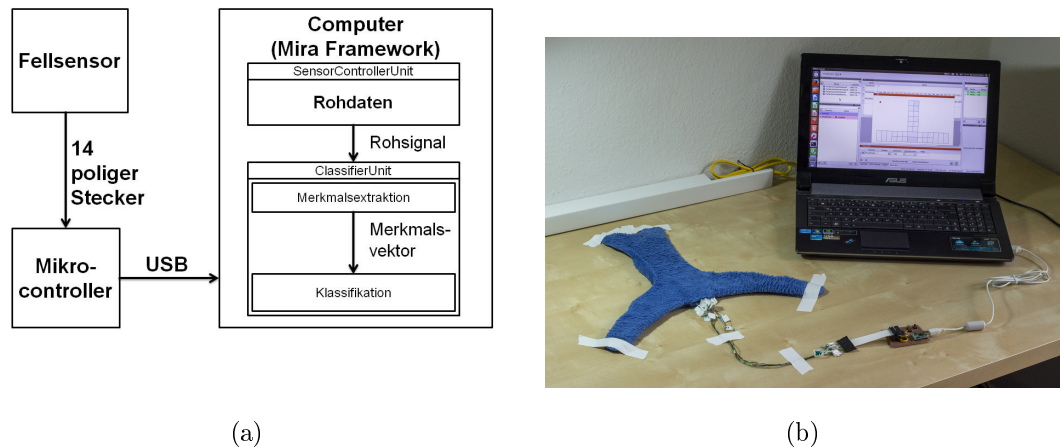
### 5.1 Versuchsaufbau

Um die Sensoren und Verfahren auf ihre Funktionsfähigkeit zu prüfen, wurde der folgende Versuchsaufbau genutzt. Wie in Abb. 5.1(a) und 5.1(b) zu sehen, ist der Fellsensor mit einem Mikrocontroller verbunden und dieser wiederum mit einem Computer, auf dem das Roboter-Framework „MIRA“ [EINHORN et al., 2012] installiert ist.

Der Mikrocontroller ist dafür verantwortlich, die Werte der Messkanäle aufzunehmen und den Wert mit der Nummer des Kanals an den Computer weiterzugeben. Dort werden die Daten von einer „Unit“ in MIRA empfangen. Diese sorgt zum einen für die Synchronisierung mit dem Mikrocontroller und zum anderen sammelt die Unit alle Messwerte in aufsteigender Reihenfolge der Kanäle. Wenn für jeden Kanal ein Messwert aufgenommen wurde, so wird ein Vektor mit diesen Daten in einen MIRA „Channel“ geschrieben, damit diese dann im nächsten Schritt weiterverarbeitet werden. Diese Weiterverarbeitung findet in einem MIRA „MicroUnit“ statt, das immer durch neue Daten im Channel aufgerufen wird. Während des Aufrufs werden die ausgewählten Merkmale extrahiert, mit den Verfahren aus Kap. 4 aufbereitet und schließlich wird mit Hilfe des Klassifikators ein Ergebnis erzeugt. Das Ergebnis wird dann wiederum auf einem Channel veröffentlicht, so dass eine weitere Nutzung der Ergebnisse möglich ist.

Die Erstellung der Trainingsdaten geschah dadurch, dass die unskalierten Merkmale

---



**Abbildung 5.1:** Blockschaftbild und Foto des Versuchsaufbaus

*Auf den Bildern ist der Versuchsaufbau abgebildet, dabei wird in (a) das Blockschaftbild und in (b) ein Foto gezeigt.*

aus einem MIRA Channel aufgenommen wurden. Danach wurden die aufgenommenen Daten mit Hilfe von Matlab [MATLAB, 2011] gelabelt und damit eine SVM trainiert, die darüber hinaus auch eine Datei mit Skalierungsinformationen für die Merkmale erstellt.

## 5.2 Ergebnisse

### 5.2.1 Garnsensor

Mit dem Garnsensor konnte das Ziel der Berührungsklassifikation nicht erreicht werden. Bereits beim Sensor selbst treten Probleme auf, die die spätere Klassifikation behindern. Aufgrund der Konstruktion verhält sich der Sensor wie eine einzelne Fläche, was sowohl zu erwünschten (Position in der Fläche) wie auch zu unerwünschten (Crosstalk) Nebeneffekten führt. Des Weiteren liefert der Sensor auch keine direkten Informationen über den Ort der Berührung und den Druck, der auf den Sensor ausgeübt wird. Diese zwei Informationen liegen abstrakt vor und müssen durch Kombination verschiedener Kanäle gewonnen werden. Auch führen „Hysterse“-Effekte zu starken

Nachteilen. Dabei wird nach einer Interaktion nicht derselbe Ausgangszustand erreicht wie vor der Interaktion, wodurch sich das Verhalten bei einer Berührung verändern kann (Siehe Kap. 3.1.2).

Diese Probleme führen weiterhin dazu, dass sich die Merkmalsextraktion als schwierig darstellt. Die Merkmale, die extrahiert wurden, haben weiterhin einen starken Bezug zu den einzelnen Kanälen. Daraus folgt, dass das Abstraktionslevel der Merkmale sehr gering ist. Deshalb werden zu einer guten Klassifikation sehr viele Testbeispiele benötigt, da durch die starke Abhängigkeit zu den Kanälen möglichst viele Varianten der Berührungen erfasst werden müssen. Um aussagekräftige Merkmale zu erhalten, wurde auch eine Analyse im Frequenzbereich durchgeführt, die jedoch keine verwertbaren Ergebnisse lieferte.

Um die Einflüsse dieser Probleme zu verringern, wurden mehrere Maßnahmen durchgeführt. Zu diesen Maßnahmen gehört die Veränderung der Zeitspanne beim Zeitfenster. Dort wurde sowohl mit einer Sekunde, als auch mit zwei Sekunden gearbeitet. Je größer diese Zeitspanne wird, desto größer wird auch die Verzögerung bis zur Klassifikation. Weiterhin wurde auch bei der Markierung der Trainingsdaten verschiedene Ansätze getestet. Zum einen wurde bei der Auswahl des Zeitslots variiert, der als positiv bewertet werden soll, zum anderen wurde auch die Größe des zu markierenden Bereichs verändert. Eine Variation der Zeitslots stellt immer einen Kompromiss dar. Je weiter der Bereich in Richtung der jüngsten Daten gelegt wird, desto weniger Informationen über den weiteren Verlauf sind bekannt und je geringer ist die Verzögerung bei der Ausgabe. Das entgegengesetzte Verhalten ergibt sich, wenn der Bereich in Richtung der ältesten Daten verschoben wird. Dabei gehen dann Informationen über den vergangenen Verlauf des Signals verloren, so dass nicht entschieden werden kann, ob der aktuelle Abschnitt eines Signals nicht Teil eines länger andauernden Signals war. Bei der Markierung wurde ebenfalls ein Ansatz zur Verbesserung der Generalisierung eingefügt. Dabei wurde um einen positiv bewerteten Bereich ein Pufferbereich festgelegt, deren Daten aus dem Trainingsdatensatz gelöscht wird. Da ein Signalverlauf zeitlich nacheinander in jedem Slot abgebildet wird, soll so vermieden werden, dass beim Lernen zu stark auswendig gelernt wird. Anstatt nur auf den ausgewählten Zeitslot fokussiert zu werden, kann so auch das Signal in anderen Zeitslots erkannt werden.

---

Aufgrund der wenig positiven Resultate wurden die extrahierten Merkmale auch mit Techniken der Merkmalsselektion untersucht. Dabei wurde „Mutual Information Feature Selection“ eingesetzt, wobei dies nicht zu einer Verbesserung der Leistung führte, da nur eine geringe Anzahl an Merkmalen als weniger wichtig deklariert wurden.

Werden die SVM im 1-vs-all Prinzip mit den gelabelten Daten trainiert, so kann keine erfolgreiche Klassifikation verzeichnet werden. Werden die SVM nur mit ihren eigenen Daten und dem Ausgangsmuster in der Ruhephase trainiert, so schlagen bei Tests alle SVMs an. Daraus lässt sich schließen, dass sich die ausgewählten Berührungen in den extrahierten Merkmalen sehr ähneln und somit die Unterscheidung dieser Klassen nicht möglich ist.

Als Resultat dieser Probleme stellt sich der Garnsensor als ungeeignet dar.

### 5.2.2 Sensor mit silberbeschichteten Stoffen

Im Gegensatz zum ersten Sensor aus leitfähigem Garn verfügt der Sensor mit silberbeschichteten Flächen über herausragende Eigenschaften. Aufgrund der Konstruktion in einer Matrix-Form lassen sich gute Werte über den Druck gewinnen und den einzelnen Flächen zuordnen. Die gesamte Fläche wurde in 32 kleine Flächen unterteilt, die hohe Unabhängigkeit voneinander besitzen. Durch den Aufbau des Sensors sind bei einigen Flächen „Crosstalk“-Effekt aufgetreten. Da der Einfluss des Effekts sehr gering ist, kann dieser vernachlässigt werden. Des Weiteren erreicht der Sensor nach einer Interaktion sofort einen stabilen Ausgangszustand.

Durch diese Ausgangslage war es möglich, Merkmale mit einem hohen Abstraktionsgrad zu gewinnen. So haben die einzelnen, genauen Positionen und Druckwerte keinen direkten Einfluss bei der Klassifikation. Dies führt zu guten Eigenschaften bei der Generalisierung der Merkmale, da nur die Muster verglichen werden und diese auf den verschiedenen Bereichen des Sensors ähnlich sind. Idealerweise eignet sich dazu eine konvexe Fläche, die über gleich große Sensorflächen verfügt und wie eine Matrix aufgebaut ist.

Mitunter haben auch bei diesem Sensor verschiedenen Berührungen in einigen Fällen ähnliche Muster in den Merkmalen, so dass die einzelne SVM in diesem Grenzfall

---

durch das 1-vs-all Prinzip positive Klassifizierung durchgeführt hat. Behoben wurde dieser Zustand durch die Nutzung einer Mehrklassen-SVM. Dadurch entsteht zwar die Gefahr, dass Klassen falsch klassifiziert werden, aber dies lässt sich durch entsprechend viele Trainingsdaten minimieren. Außerdem bedeutet eine SVM gegenüber mehreren SVMs weniger Berechnungsaufwand, wodurch sich die Anzahl an Klassen besser skalieren lässt.

Bei einem allgemeinen Funktionstest mit der gleichen Testperson konnten zufriedenstellende Ergebnisse erreicht werden. Aufgrund des geringen Testumfangs sieht sich diese Arbeit als Machbarkeitsstudie.

Mit dem Sensor aus silberbeschichteten Stoffen und der SVM konnte eine Vereinigung von Berührungsklassifikation und textilen Sensoren erreicht werden.

---





# Kapitel 6

## Zusammenfassung und Ausblick

### 6.1 Zusammenfassung

Das Ziel der Klassifikation von Berührungsmustern durch textile Sensoren ist erreicht worden. Damit konnte eine Verbindung aus den verschiedenen recherchierten Artikeln hergestellt werden.

Der Sensor aus leitfähigem Garn stellt sich jedoch als ungeeignet heraus. Der Hauptgrund dafür ist das Verhalten als einzige Sensorfläche, die viele Nebeneffekte hervorruft. Der zweite Entwurf mit silberbeschichteten Stoffen, der aufgrund der Recherchearbeiten entwickelt wurde, hat in ersten Tests gute Ergebnisse geliefert. Damit stellt sich diese Arbeit als Machbarkeitsstudie für die Vereinigung von textilen haptischen Sensoren und der Klassifikation von Berührungen dar.

### 6.2 Ausblick

Um die Einsatzreife des zweiten Sensors zu erreichen, muss die Hardwaregrundlage noch einmal überarbeitet und verfeinert werden.

Um die Funktionsfähigkeit zu verbessern, bietet sich eine rechteckige Grundform an, da die hier gewählte Form (Siehe Abb. 3.13) am Übergang der äußeren Stücke zum mittleren Stück zu Fehlklassifikationen neigen kann, wenn die Hand nicht vollständig auf der Sensorfläche platziert wird. Des Weiteren sollten die einzelnen Sensorflächen

---

eine genormte, gleiche Größe haben, damit die Abstraktion aus den Daten die korrekten Gegebenheiten widerspiegelt und nicht durch unterschiedliche Größen der Flächen verzerrt wird. Sollte dies nicht möglich sein, muss untersucht werden, inwieweit sich dies negativ auswirkt und ob diese Abweichung kompensiert werden kann.

Die Verkabelung und Beschaltung muss ebenfalls noch einmal untersucht werden. Wie im Kap. 3.2 beschrieben, verursacht das Parallelschalten einiger Flächen unerwünschte Crosstalk Effekte. Sollte die Reihenschaltung tatsächlich weniger Störungen verursachen, so könnte man mit Hilfe der 14 Anschluss Pins insgesamt 48 Sensorflächen nutzen. Dabei sollte dann gegebenenfalls leistungsfähigere Hardware genutzt werden, um eine ausreichende Übertragungsrate zu gewährleisten.

Abschließend wäre eine Testserie mit verschiedenen Testpersonen durchzuführen.

---

# Anhang A

## Mikrocontroller

In diesem Anhang zum Mikrocontroller wird kurz beschrieben, wie die verschiedenen Softwareversionen des auf dem Mikrocontroller laufenden Programms auf den Controller aufgespielt werden. Weiterhin wird auch die genutzte Software und das genutzte Betriebssystem dafür erwähnt.

genutzte Software beim Mikrocontroller		
Betriebssystem	Software	Einsatzgebiet
Windows 7	Atmel Studio 6 (Version 6.1.2730) SP 2	Programmierung der Software für Mikrocontroller
Windows 7	chip45boot2 GUI (Version 1.13)	Aufspielen der Software auf den Mikrocontroller

**Tabelle A.1:** In dieser Tabelle ist die genutzte Software aufgelistet, die für die Arbeiten am Mikrocontroller genutzt wurde.

---

## A.1 Anleitung zum Aufspielen der Software

1. Schließen Sie den Mikrocontroller an den Zielrechner an.
2. Öffnen Sie chip45boot2 GUI.
3. Wählen Sie den korrekten Port im Fenster „Select COM Port“ aus (z.B. COM4) und stellen Sie ggf. die Baudrate im Fenster „Baudrate“ auf 19200.
4. Drücken Sie auf den Reset Button des Mikrocontrollers und darauf folgend auf „Connect to Bootloader“ in chip45boot2.
5. Wählen Sie das gewünschte Flash oder Eeprom Hexfile mit „Select Flash Hexfile“ bzw. „Select Eeprom Hexfile“ aus.

## A.2 Pfade zu den Hex- und Eeprom-Dateien:

<Version>\Release\Fellsensor.hex

<Version>\Release\Fellsensor.eep

### Versionen:

1. **Fellsensor \_Garn \_ohneWiderstaenden**

Erste Version der Software für den Sensor aus leitfähigem Garn.

2. **Fellsensor \_Garn \_mitWiderstaenden**

Zweite Version der Software für den Sensor aus leitfähigem Garn, bei dem am Mikrocontroller Widerstände angelötet wurden.

3. **Fellsensor \_Matrix**

Erste Version der Software für den Sensor mit silberbeschichteten Stoffen.

---

## Anhang B

### MIRA Programme

In diesem Anhang werden verschiedene Aspekte zu der entwickelten Software erklärt. Am Anfang wird eine kurze Übersicht über die verwendete Software gegeben. Diese Übersicht ist in Tab. B.1 gegeben. Danach folgen die Erklärungen zu den Konfigurationsdateien „FurSensor.xml“ und „SVMTrainer.xml“. Im Anschluss daran wird der Ablauf zur Verarbeitung von Berührungsmustern geschildert, beginnend bei der Aufnahme mit Hilfe von MIRA Center, gefolgt von der Markierung der Klassen mit Hilfe von Matlab und schließlich die Erstellung einer SVM aus den gelabelten Daten.

genutzte Software beim Mira Framework		
Betriebssystem	Software	Einsatzgebiet
Ubuntu 12.04	Eclipse IDE for C/C++ Developers Kepler (SR 2)	Programmierung der Software für Mira
Windows 7	Matlab R2011b (7.13) 64-bit	Labeln und Auswerten der Daten
Windows 7	JSONlab 0.9.9 (Optimus - beta)	Matlab Toolbox zum Extrahieren von JSON Dateien

**Tabelle B.1:** In dieser Tabelle ist die genutzte Software aufgelistet, die für die Arbeiten am Mira Framework und bei der Klassifikation genutzt wurde.

---

```

<root>
  <unit id="MyFurSensor" class="fursensor::FurSensorController" >
    <Device>/dev/ttyUSB0</Device>
  </unit>
  <unit id="MyTouchClassifier" class="fursensor::SVMTouchClassifier" >
    <FeatureScalingParmeter>domains/FurSensor/etc/featureScaling.xml</FeatureScalingParmeter>
    <RawScalingParmeter>domains/FurSensor/etc/rawScaling.xml</RawScalingParmeter>
    <!-- if RawScalingParameter does not exist, LiveScaling will be automatically set to true! -->
    <LiveScaling>true</LiveScaling>
    <SVMClassifierFileName>domains/FurSensor/etc/MulticlassSVM</SVMClassifierFileName>
    <!-- if SVM file does not exist EnableClassification will be automatically set to false! -->
    <EnableClassification>true</EnableClassification>
  </unit>
</root>

```

Abbildung B.1: FurSensor.xml

*Enthält Einstellungen und Parameter für das Ausführen der Software.*

## B.1 Erklärung FurSensor.xml

Device	enthält den Pfad zu dem Verzeichnis, an dem der Mikrocontroller verbunden ist
FeatureScalingParmeter	enthält den Pfad zur Datei, die die Skalierungsparameter für die Merkmale enthält
RawScalingParmeter	enthält den Pfad zur Datei, die die Skalierungsparameter für die Rohwerte enthält
LiveScaling	true: während der Laufzeit werden die Skalierungsparameter der Rohwerte angepasst und gespeichert. Existiert keine Datei „RawScaling-Parameter“, so wird diese Variable automatisch auf true gesetzt
SVMClassifierFileName	enthält den Pfad zur Datei, in der die trainierte SVM gespeichert ist
EnableClassification	true: lädt den Klassifikator und veröffentlicht den Kanal für die Klassifikationen. Wird die Datei aus „SVMClassifierFileName“ nicht gefunden, so wird diese Variable automatisch auf false gesetzt

```

<root>
  <unit id="MySVMTrainer" class="fursensor::SVMTrainer">
    <InputLabelLabeledMulticlassFileName>domains/FurSensor/etc/LabeledMulticlassData</InputLabelLabeledMulticlassFileName>
    <OutputMulticlassSVMFileName>domains/FurSensor/etc/MulticlassSVM</OutputMulticlassSVMFileName>
    <OutputScalingFileName>domains/FurSensor/etc/featureScaling</OutputScalingFileName>
    <NumberOfFeatures>4</NumberOfFeatures>
    <!-- enum { C_SVC, NU_SVC, ONE_CLASS, EPSILON_SVR, NU_SVR }; /* svm_type */-->
    <SVMTType>0</SVMTType>
    <!-- enum { LINEAR, POLY, RBF, SIGMOID, PRECOMPUTED }; /* kernel_type */-->
    <SVMKernel>2</SVMKernel>
    <!-- for poly -->
    <SVMDegree>3</SVMDegree>
  </unit>
</root>

```

Abbildung B.2: SVMTrainer.xml

*Enthält Einstellung und Parameter für das Ausführen der Software.*

## B.2 Erklärung SVMTrainer.xml

InputLabelLabeledMulticlassFileName	enthält den Pfad zur Datei, die die gelabelten Daten enthält
OutputMulticlassSVMFileName	enthält den Pfad und den Namen der Datei, die als Ausgabe für die trainierte SVM genutzt wird
OutputScalingFileName	enthält den Pfad und den Namen der Datei, die als Ausgabe für die Skalierungsparameter der Merkmale genutzt wird
NumberOfFeatures	Anzahl der Klassen wird benötigt, um die Skalierungsparameter korrekt zu initialisieren
SVMTType	der Typ der SVM, der genutzt wird
SVMKernel	die Kernel-Funktion, die von der SVM genutzt wird
SVMDegree	Steilheit der Kernel-Funktion

## B.3 Anleitung zum Aufnehmen von Daten

1. Wechseln Sie in den Ordner „tactilesensor“.
2. Starten Sie die Software mit Hilfe von MIRA Center:

```
miracenter domains/FurSensor/etc/FurSensor.xml
```

3. Wählen Sie den Reiter „Recorder“ aus, der sich mittig und am unteren Rand des MIRA Center befindet.
  4. Ziehen Sie per Drag-and-Drop den Kanal „FurSensorFeatureChannel“ in das Fenster des Recorders.
  5. Klicken Sie auf den Aufnahme-Button (roter Kreis), wählen Sie dann ein Verzeichnis und einen Dateinamen aus.
  6. Darauf startet sofort die Aufnahme.
  7. Zum Stoppen klicken Sie auf den Stopp-Button (rotes Viereck). Diesen Prozess müssen Sie für jede Berührungsklasse separat durchführen.
-



## B.4 Anleitung zum Labeln einer <name>.tape Datei

1. Zuerst müssen Sie die <name>.tape Dateien in ein anderes Format umwandeln, damit es leichter von Matlab verarbeitet werden kann. Dies geschieht mit folgendem Befehl: (Siehe auch Abb. B.1)

```
miratape writeJSON <tapeName>.tape <filename> --matlab
```

<fileName>.tape ist die zuvor aufgenommene Datei und <filename> ist die Zielformatdatei, die dabei erstellt wird.

2. Nun kopieren Sie die Dateien in den Basisordner, in dem das Matlab-Skript „executeAutomaticLabeling“ enthalten ist. Stellen Sie sicher, dass alle gewünschten Klassen im Ordner abgelegt sind.
3. Jetzt öffnen Sie die Datei „executeAutomaticLabeling.m“ und tragen alle Dateinamen in das „nameArray“ ein.
4. Danach führen Sie das Skript aus und eine neue Datei wird erstellt (Default:LabeledMulticlassData). Diese Datei enthält nun alle Klassen mit den entsprechenden Labeln. Die Label werden entsprechend der Reihenfolge im „nameArray“ vergeben und beginnen bei 1.

## B.5 Anleitung zum Erstellen einer SVM

1. Kopieren Sie die zuvor erstellten Datei wieder in das System, in dem MIRA läuft.
2. Öffnen Sie folgende Datei:

```
domains/FurSensor/etc/SVMTrainer.xml
```

Geben Sie nun die Adresse der Datei in die Tags zu `<InputLabelMulticlassFileName>` ein. Sie können nun auch den Namen der Ausgabedatei (`<OutputMulticlassSVMFileName>`) und der Datei mit Skalierungsinformationen (`<OutputScalingFileName>`) anpassen.

3. Stellen Sie sicher, dass die korrekte Anzahl an Merkmalen in `<NumberOfFeatures>` ausgewählt ist. Dieser Wert beeinflusst die Erstellung der Skalierungsdatei, die später bei der Klassifikation benötigt wird.
4. Mit den weiteren drei Parametern können Sie die Einstellungen der SVM anpassen.
5. Führen Sie das Programm nun mit folgendem Befehl aus: (Siehe auch Abb. B.2)

```
mira domains/FurSensor/etc/SVMTrainer.xml
```

6. Es wurden nun die zwei neuen Dateien erstellt. Damit Sie diese nun nutzen können, öffnen Sie:

```
domains/FurSensor/etc/FurSensor.xml
```

Geben Sie nun den Pfad inklusive Dateinamen in die Felder `<FeatureScalingParameter>` und `<SVMClassifierFileName>` ein.

7. Um das Programm inklusive Klassifikation zu starten, führen sie erneut den folgenden Befehl aus:

```
miracenter domains/FurSensor/etc/FurSensor.xml
```

---

```

<root>
  <unit id="MySVMTrainer" class="fursensor::SVMTrainer" >
    <InputLabelLabeledMulticlassFileNameVector>
      <item>domains/FurSensor/etc/klaps</item>
      <item>domains/FurSensor/etc/kraulen</item>
      <item>domains/FurSensor/etc/schlag</item>
      <item>domains/FurSensor/etc/streicheln</item>
    </InputLabelLabeledMulticlassFileNameVector>
    <OutputSingleclassSVMFileNameVector>
      <item>domains/FurSensor/etc/klaps.xml</item>
      <item>domains/FurSensor/etc/kraulen.xml</item>
      <item>domains/FurSensor/etc/schlag.xml</item>
      <item>domains/FurSensor/etc/streicheln.xml</item>
    </OutputSingleclassSVMFileNameVector>
    <OutputScalingFileName>domains/FurSensor/etc/featureScaling</OutputScalingFileName>
    <NumberOfFeatures>280</NumberOfFeatures>
    <!-- enum { C_SVC, NU_SVC, ONE_CLASS, EPSILON_SVR, NU_SVR }; /* svm_type */-->
    <SVMType>0</SVMType>
    <!-- enum { LINEAR, POLY, RBF, SIGMOID, PRECOMPUTED }; /* kernel_type */-->
    <SVMKernel>2</SVMKernel>
    <!-- for poly -->
    <SVMDegree>3</SVMDegree>
  </unit>
</root>

```

Abbildung B.3: SVMTrainer.xml für Garnsensor

Enthält ein Beispiel für die XML Array Darstellung.

## B.6 Besonderheit beim Garnsensor

- In Matlab nutzen Sie das Skript „A\_convertJSONtoCSV“ und danach müssen Sie jede erstellte Datei manuell mit dem Skript „B\_labelData“ markieren.
- Im Skript „B\_labelData“ wird dazu ein Vector „indices“ bereit gestellt, mit dem Sie an Hand der Visualisierung die korrekten Punkte markieren müssen. Dazu gibt es eine Option, die genau den dritten Slot des Zeitfenster anzeigt, um einfacher markieren zu können. Sie haben dann noch die Auswahl, die ungeschnittene oder die geschnittene Variante der Daten zu speichern.
- Unter Ubuntu bzw. in den XML Konfigurationsdateien müssen Sie die neu erstellten Dateien als Array importieren. In Abb. B.3 ist das Vorgehen zu sehen. Es wird entsprechend der Eingabe-/Ausgabepfad jeweils in <item> Tags hinzugefügt.
- Da auch beim SVMTrainer wieder vier Dateien entstehen, müssen Sie diese in der gleichen Art und Weise in der Datei „FurSensor.xml“ hinzufügen.



# Abbildungsverzeichnis

1.1	Blauer Fellsensor auf dem Roboter . . . . .	2
2.1	Querschnitt des textilen Ganzkörpersensors . . . . .	5
2.2	Querschnitt des textilen Sensorhandschuhs . . . . .	7
2.3	Aufbau der Sensorfläche und die Beschaltungstabelle . . . . .	8
2.4	Foliensensor und Grafik für die Verteilung der Muster bei KNN . . . . .	12
2.5	Versuchsaufbau und PIFACT Terme für das neuronale Netz . . . . .	13
2.6	extrahierte Merkmale für das neuronale Netz . . . . .	13
2.7	Querschnitt des Sensors mit Photointerrupter für die SVM . . . . .	15
2.8	Vergleich SVM und KNN . . . . .	15
3.1	Blockbild des Versuchsaufbaus mit Markierung für Sensor . . . . .	19
3.2	Querschnitt des Fellsensors aus gestricktem, leitfähigem Garn . . . . .	21
3.3	Foto des leitfähigen Garn . . . . .	21
3.4	Draufsicht auf den internen Aufbau des Sensors . . . . .	22
3.5	Ersatzschaltbild des Messansatzes . . . . .	23
3.6	Rohdaten des Fellsensor (gestricktes Garn) V1 . . . . .	24
3.7	Einschwingzeit des Fellsensors (gestricktes Garn) . . . . .	25
3.8	Fehlerquelle in den Rohdaten des Fellsensors (gestricktes Garn) . . . . .	26
3.9	Rohdaten eines Kanals des Fellsensors (gestricktes Garn) . . . . .	27
3.10	Ersatzschaltbild des Messansatzes . . . . .	28
3.11	Rohdaten des Fellsensor (gestricktes Garn) V2 . . . . .	29
3.12	Querschnitt des Fellsensors aus silberbeschichteten Stoffen . . . . .	30
3.13	Draufsicht und Foto des Sensors mit silberbeschichteten Stoffen . . . . .	31

---

3.14 Rohdaten des Fellsensors (silberbeschichtete Stoffe) . . . . .	34
4.1 Blockbild des Versuchsaufbaus mit Markierung für MicroUnit . . . . .	36
4.2 Schema des Zeitfensters . . . . .	37
4.3 Beispiel Rohdaten mit zwei Zeitfenstern . . . . .	38
4.4 maximale Abweichung vom Mittelwert . . . . .	39
4.5 Standardabweichung . . . . .	40
4.6 Vorzeichenwechsel vom Mittelwert . . . . .	40
4.7 gelabelte Daten mit und ohne Pufferbereich . . . . .	42
4.8 Blockbild des Sample-and-Hold-Ansatzes . . . . .	44
4.9 Pfadlängen aller vier Klassen nebeneinander . . . . .	46
4.10 Maximale Abweichung vom ersten gewichteten Mittelpunkt . . . . .	46
4.11 Maximale Summe der Druckwerte der aktivierten Flächen zu einem Zeitpunkt . . . . .	47
4.12 Die größte Anzahl an aktivierten Flächen zu einem Zeitpunkt . . . . .	47
4.13 Foto einer Streichelbewegung . . . . .	48
5.1 Blockschaltbild und Foto des Versuchsaufbaus . . . . .	52
B.1 FurSensor.xml . . . . .	62
B.2 SVMTrainer.xml . . . . .	63
B.3 SVMTrainer.xml für Garnsensor . . . . .	67

---

## Literaturverzeichnis

- [BUSCHER et al., 2012] BUSCHER, G., R. KOIVA, C. SCHURMANN, R. HASCHKE und H. RITTER (2012). *Tactile dataglove with fabric-based sensors*. In: *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, S. 204–209.
- [CHANG und LIN, 2011] CHANG, CHIH-CHUNG und C.-J. LIN (2011). *LIBSVM: A library for support vector machines*. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [COONEY et al., 2012] COONEY, M.D., S. NISHIO und H. ISHIGURO (2012). *Recognizing affection for a touch-based interaction with a humanoid robot*. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, S. 1420–1427.
- [EINHORN et al., 2012] EINHORN, E., T. LANGNER, R. STRICKER, C. MARTIN und H. GROSS (2012). *MIRA - middleware for robotic applications*. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, S. 2591–2598.
- [GRIES, 2013] GRIES, VANESSA (2013). *Haptische Interfaces für Mensch Roboter Interaktion*. Hauptseminar im Fachgebiet Neuroinformatik und Kognitive Robotik TU-Ilmenau.
- [GWILLIAM et al., 2010] GWILLIAM, J.C., Z. PEZZEMENTI, E. JANTHO, A. OKAMURA und S. HSIAO (2010). *Human vs. robotic tactile sensing: Detecting lumps in soft tissue*. In: *Haptics Symposium, 2010 IEEE*, S. 21–28.
-

- [INABA et al., 1996] INABA, M., Y. HOSHINO, K. NAGASAKA, T. NINOMIYA, S. KAGAMI und H. INOUE (1996). *A full-body tactile sensor suit using electrically conductive fabric and strings*. In: *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, Bd. 2, S. 450–457 vol.2.
- [IWATA und SUGANO, 2005] IWATA, H. und S. SUGANO (2005). *Human-robot-contact-state identification based on tactile recognition*. *Industrial Electronics, IEEE Transactions on*, 52(6):1468–1477.
- [MATLAB, 2011] MATLAB (2011). *version 7.13 (R2010a)*. The MathWorks Inc., Natick, Massachusetts.
- [NAYA et al., 1999] NAYA, F., J. YAMATO und K. SHINOZAWA (1999). *Recognizing human touching behaviors using a haptic interface for a pet-robot*. In: *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, Bd. 2, S. 1030–1034 vol.2.
- [NIKR] NIKR. Fachgebiet Neuroinformatik und Kognitive Robotik TU-Ilmenau. <http://www.tu-ilmenau.de/neurob/>.
- [PUANGMALI et al., 2008] PUANGMALI, PINYO, K. ALTHOEFER, L. SENEVIRATNE, D. MURPHY und P. DASGUPTA (2008). *State-of-the-Art in Force and Tactile Sensing for Minimally Invasive Surgery*. *Sensors Journal, IEEE*, 8(4):371–381.
- [RUAN et al., 2010] RUAN, JUAN-YAO, P.-P. CHAO und W.-D. CHEN (2010). *A multi-touch interface circuit for a large-sized capacitive touch panel*. In: *Sensors, 2010 IEEE*, S. 309–314.
- [SCHMEDER und FREED, 2010] SCHMEDER, ANDREW und A. FREED (2010). *Support Vector Machine Learning for Gesture Signal Estimation with a Piezo Resistive Fabric Touch Surface*. In: *NIME*, Sydney, Australia. Draft paper in submission. Do not redistribute without permission.
-